U.S. SENATOR ELECTION HISTORY – AND ANDROID APPLICATION

_____

A Thesis

Presented to the

Faculty of

San Diego State University

_____

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

_____

by

Lavanya Vangalkrishnakumar

Spring 2013

SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Lavanya Vangalkrishnakumar:

U.S. Senator Election History – And Android Application

Carl Eckberg, Chair
Department of Computer Science

William Root
Department of Computer Science

Michael O'Sullivan
Department of Mathematics and Statistics

4-17-13

Approval Date

Copyright © 2013
by
Lavanya Vangalkrishnakumar
All Rights Reserved

DEDICATION

I dedicate this thesis to my husband Dr. Ravikiran Gopalan. I give my deepest expression of love and appreciation for the encouragement that you gave me and the sacrifices you made for me during my graduate program. Thank you for all the motivation you had given to pursue my masters and for your company during all the late nights I was working to get my thesis completed.

ABSTRACT OF THE THESIS

U.S. Senator Election History - An Android Application
by
Lavanya Vangalkrishnakumar
Master of Science in Computer Science
San Diego State University, 2013

The United States current senate composition for the 113th congress totals to 53 members of the Democratic party, 45 members of the Republican part and two independent members out of the 100 members. Senators are elected by their state as a whole. The idea is two senators per state regardless of population. Originally, senators were selected by the state legislatures, not by popular elections. By the early years of the 20th century, the legislatures of as many as 29 states had provided for popular election of senators.

Senators serves for a staggered term of six years. This is so that approximately one-third of the seats are up for election every two years. The three classes of United States Senators are made up of 33 or 34 Senate seats each. Class 1 and Class 2 consist of 33 senators and class 3 has 34 senators. Should a 51st state be admitted, it would receive senators in Classes 1 and 2, at which point all three Classes would have 34 senators. Each state belongs to two of the three classes. The main purpose of the classes in the senate is to determine which senate seats will be up for election in a given year. Popular election to the Senate was standardized nationally in 1913 by the 17th Amendment.

An elected senator serves for a staggered tem of six years. This is so that approximately one- third of the seats are up for election every two years. The three classes of United States Senators are made up of 33 or 34 Senate seats each. Class 1 and class 2 consist of 33 senators and class 3 has 34 senators. Should a 51st state be admitted, it would receive senators in classes 1 and 2, at which point all three classes would have 34 senators. Each state belongs to two of the three classes. The main purpose of the classes in the senate is to determine which Senate seats will be up for election in a given year. Popular election to the senate was standardized nationally in 1913 by the 17th amendment.

While there is a lot of information available about the senate, in this fast paced mobile world, we would like to have the information readily available any place any time. The main focus here is to provide a quick but a detailed summary about the senate and its election trends. The application is developed as an Android application for mobile phones /tablets. The user can look up senate/senator facts on the fly using their mobiles/tablets.

The application is intended for self-study by students at any level, K-12 or college, e.g. in history teachers can give assignments based on the political science application by including questions to be answered by a student, after exploring the app.

The application can be downloaded for any device that runs Android version 2.3 and above. Due to extensive use of a database all the information necessary for the application is self contained. A user needs to have JRE installed on his/her system. The android application is written in the Java programming language and developed using

Android SDK and other necessary plugins in Eclipse IDE.

TABLE OF CONTENTS

PAGE

ABSTRACT v
LIST OF FIGURES ix
ACKNOWLEDGEMENTS xi
CHAPTER
1 INTRODUCTION 1
2 SENATOR 3
2.1 Senator Overview 3
2.1.1 About ‘ Senate ’ 3
2.1.2 Composition and Election Term 3
2.2 Senator Seniority 4
2.2.1 Deterministic Factors for Seniority 4
2.2.2 Seniority Based Benefit 4
2.3 Senator Classes Based On State 5
2.3.1 Evolution of Class Based Senate Election 5
2.3.2 List of States Based on Senate Classes 6
2.4 Powers of Senator over House 6
3 TECHNOLOGY 8
3.1 Android Overview 8
3.1.1 About Android 9
3.1.2 Name – ‘Android’ 9
3.1.3 Evolution of Android 10
3.1.4 Phone Manufactures that Use Android 10
3.1.5 Android Version 10
3.1.6 Android Architecture 11
3.1.7 Dalvik Virtual Machine 11
3.2 Android Software Development Kit 11
3.2.1 Overcome of Traditionally Challenges 12

3.2.2 Android Emulator 13
3.2.3 Android Debug Bridge 13
3.2.4 Android Development Tools (ADT) 14
3.3 Eclipse 14
3.4 SQLite Overview 15
4 THE REQUIREMENTS 16
4.1 Data Requirements 16
4.2 Platform Requirements 17
4.3 Functional Requirements 17
5 ANALYSIS 19
6 DATA COLLECTION AND PREPARATION OF DATA 21
7 APPLICATION DEVELOPMENT WITH FEATURES 26
7.1 Implementaion 26
7.2 Main Application Screen Flow - Screens 26
7.2.1 Main Screen 26
7.2.2 Region Selection Screen 31
7.2.3 State Selection Screen 34
7.2.4 Tutorial Screen 34
7.2.5 Quiz Screen 35
7.2.6 Advanced Search Screen 38
8 CONCULSION AND OBSTRACLES 40
9 FUTURE WORK 41
REFERENCES 42
APPENDIX
A USER MANUAL 43
B APPLICATION CODE 53

LIST OF FIGURES

PAGE

Figure 2.1. Senate classes in each state. 7
Figure 3.1. Android logo. 9
Figure 3.2. Android version. 11
Figure 3.3. Android architecture 12
Figure 3.4. Simplified android software view. 13
Figure 3.5. Android emulator 14
Figure 6.1. State based senator URL's. 22
Figure 6.2. California state senator text file. 23
Figure 6.3. Senator information for state of Washington. 23
Figure 6.4 Senator information for state of Montana. 24
Figure 6.5. Senator information for state of Idaho. 24
Figure 6.6. Senator information for state of Wyoming. 25
Figure 7.1. Senator application project structure in eclipse. 27
Figure 7.2. Main US map layout structure. 28
Figure 7.3. Tutorial screen layout structure. 34
Figure 7.4. Quiz screen layout structure. 35
Figure 7.5. Advanced search screen layout structure. 38
Figure 9.1. Interactive instruction sample. 41
Figure A.1. Screen shot of SDK starter package includes the Android SDK and AVD Manager. 44
Figure A.2. Steps to find and execute SDK manager. 45
Figure A.3. Android platform version choice. 45
Figure A.4. Screen shot of how to open Eclipse. 46
Figure A.5. Screen shot of the dialog to type 'ADT Plugin'. 46
Figure A.6. Screenshot of 'developer tools' option. 47
Figure A.7. Screenshot to select Android from the panel on the left. 48
Figure A.8. Screen shot in Eclipse, navigate to Window --> AVD Manager. 49
Figure A.9. Screenshot how to create AVD. 49

Figure A.10. Show the screenshot of run configuration option. 50
Figure A.11. Shows the screenshot of emulator/device option. 51
Figure A.12. Shows the screen shot of emulator option. 51
Figure A.13. Screen shot to run on device. 52
Figure B.1. Shows main.xml graphical layout. 58
Figure B.2. Shows westcoastus.xml graphical layout. 61
Figure B.3. Shows estus.xml graphical layout. 64
Figure B.4. Shows southus.xml graphical layout. 67
Figure B.5. Shows midwestus.xml graphical layout. 70
Figure B.6. Shows quizview.xml graphical layout. 75

ACKNOWLEDGEMENTS

I thank my advisor Dr. Carl Eckberg for giving me an opportunity to work on this thesis project and for his constant guidance, support and motivation throughout this Thesis. I would also like to thank Professor Root, William and Professor Michael O' Sullivan for being on my committee and for their help and co-operation.

CHAPTER 1

INTRODUCTION

California has two senators Diane Feinstein and Barbara Boxer, both democrats, belonging to class1 and class3, these terms are explained in Chapter 2. They have served as senators from November 10, 1992 –Present and January 3, 1993 –Present respectively. Recollecting such facts might be cumbersome for all the United States current senators nor is it easy to comprehend all the election terms and regulations associated with the senate. The application gives quick access to the history and details pertaining to the United States senate using a mobile application on android devices. The application covers all the details ranging from senates election terms and its regulation policy to the election results based on different states over about 100 years.

The purpose of this application is to provide students and teachers of political science and history departments with a mobile based application that can aid in a playful way of learning about the US senate and acquiring/refreshing their memory at any given place/time. More and more schools are moving towards mobile learning as a way to take advantage of portability and ease of use in budget. This application will offer teachers and students a more flexible approach to learning about the senate rules and their powers in the United State congress.

The users can interact with the application to get quick facts about the current and the previous senates of the U.S. They can also refresh their memory about the senate policy and powers using the tutorial feature. This application provides precise as well as a detailed levels of information about the senate on a state by state basis. There is much more users can do using this application which will be discussed in the following chapters.

The interface of the software being used to develop this application is very similar to other apps that are available in the market store. Google is the company that provides the software development kit for android apps development. The application is developed using JAVA as the programming language. ‘Eclipse’ is used as the IDE for the

development of the code. The application stores all the senator related information of each state as an internal database. SQLite is the database that is integrated along with android if any database related development needs to be done for the application. The front end user interface screens are developed using xml scripts. These layouts are then connected to the java programming language to display them on the android devices, both tablets and phones.

The thesis report has been organized into eight chapters. The first chapter gives an introduction about the thesis. The second chapter has an overview of the senate with its class system and election rules. The third chapter discusses the technologies used and available. Chapter 4 discusses platform requirements and functional requirements. In the fifth chapter, the requirements are analyzed in steps for the development of this application. In the sixth chapter has a detailed discussion of data collection and preparation of the data. In the seventh chapter, the application's features and the corresponding code is displayed. In chapter eight, future enhancements to the application are discussed.

CHAPTER 2

SENATOR

This chapter describes the overview of the senate, its responsibilities and election terms with services in detail. This chapter also gives a comparison of the two branches of the United States congress.

2.1 SENATOR OVERVIEW

The United States senate and the senate related details form the core understanding of the subject of the thesis. Senators being the core theme of this application, one must know the structure of the U.S. senate, its class and election terms, roles and responsibilities.

2.1.1 About ‘ Senate ’

The United States congress is made up of two houses of congress (a bicameral legislature). The ‘Senate’ forms the upper house of the bicameral legislature and the lower house of the bicameral legislature is referred to as “the house of representatives” found in [1]. Historically the term “Senate” means the assembly of the eldest and wiser members of the society and ruling class. The first official senate was the Roman senate of Ancient Rome as found in [2].

2.1.2 Composition and Election Term

The United States is made up of 50 states. Two senators represent each state, irrespective of the population of the state. For instance, Rhode Island being the smallest of size around ~1,045 sq miles has two senators elected for that state. On the other hand Texas being the largest state still has only two senators that are elected. The two senators from a given state each serve a staggered term of six years. The intention is to present states large in area or population, from dominating smaller states.

2.2 SENATOR SENIORITY

Each of the senators elected has a description about their seniority. Each senator is either classified as a “senior” or “junior”. Seniority in the senate plays a valuable role as the number of benefits is based on the continuous length of the duration served, along with ties broken by a series of factors

2.2.1 Deterministic Factors for Seniority

Continuous Service – Main factor

- Former Senator
- Former Representative
- Former President of the United States
- Former Vice President of the United States
- Former Cabinet member
- Former state Governor
- Population of state based on the most recent census when the senator took office
- Alphabetical by last name (in case two senators came from the same state on the same day and have identical credentials) as found in [3].

The above-mentioned factors are used to break the tie when there are two senators who have served in the same previous role, length of time in that prior office. For instance Barbara Mikulski, Richard Shelby, John McCain and Harry Reid all took office on January 3, 1987, and each had previously served in the House of Representatives. Mikulski, having served 10 years, is more senior than Shelby, who served 8. They are both more senior than McCain and Reid, who each served 4 years. McCain, however, outranks Reid because Arizona's population was higher than Nevada's as of the 1980 census as found in [3].

2.2.2 Seniority Based Benefit

The constitution does not mandate a difference in rights or power but senate rules give more power to senior senators when compared with the juniors. The ‘president pro tempore of the Senate’ is traditionally the most senior member of the majority party.

List of senior senate perks:

- Senior senators will have more power, especially within their own congressional caucuses.

- By tradition, senior senators from the president's party control federal patronage appointments in their states.
- Senators are given preferential treatment in choosing committee assignments based on seniority
- Greater seniority enables a senator to choose a desk closer to the front of the Senate Chamber.
- Senators with higher seniority may choose to move into better office space as those offices are vacated.
- Seniority determines the ranking in the ‘ United States order of precedence ’ although other factors, such as being a former President or First Lady, are involved as found in [3].

2.3 SENATOR CLASSES BASED ON STATE

Purpose: The class system is a way to determine which senate seats will be available for election in a given year. The class system is divided into three groups namely class 1, class 2 and class3. Each of the 50 states belongs to two of the class systems.

2.3.1 Evolution of Class Based Senate Election

The political leaders and statesmen who participated in the American Revolution by signing the United States Declaration of Independence, decided to stagger the elections, so that a third of the Senate was up for election every two years. The idea behind this staggered approach was to ensure stability in the senate, and encourage senators to deliberate measures over time, rather than risk a rapid turnover of the entire chamber every six years. They also wanted more frequent elections, as opposed to waiting every six years, to prevent senators from permanently combining for "sinister purposes". The three classes were specified based on article 1 of United States constitution. The plan of execution for the terms of each senate class was that the seats of the first Class shall be vacated at the expiration of the second year, of the second Class at the Expiration of the fourth year, and of the third Class at the Expiration of the sixth year, so that one third may be chosen every second year.

Upon the expiration of a senator's term of any length, someone starts a new six-year term as senator (based on election by the state legislatures until the Seventeenth Amendment required direct popular election of Senators).

2.3.2 List of States Based on Senate Classes

- Class 1

  States with a Class 1 senator: Arizona, California, Connecticut, Delaware, Florida, Hawaii, Indiana, Maine, Maryland, Massachusetts, Michigan, Minnesota, Mississippi, Missouri, Montana, Nebraska, Nevada, New Jersey, New Mexico, New York, North Dakota, Ohio, Pennsylvania, Rhode Island, Tennessee, Texas, Utah, Vermont, Virginia, Washington, West Virginia, Wisconsin, and Wyoming.

  Class 1 senators consist of the 33 senators who were last elected in November 2012 whose seats are scheduled for re-election in November 2018 as found in [4].

- Class 2

  States with a Class 2 senator: Alabama, Alaska, Arkansas, Colorado, Delaware, Georgia, Idaho, Illinois, Iowa, Kansas, Kentucky, Louisiana, Maine, Massachusetts, Michigan, Minnesota, Mississippi, Montana, Nebraska, New Hampshire, New Jersey, New Mexico, North Carolina, Oklahoma, Oregon, Rhode Island, South Carolina, South Dakota, Tennessee, Texas, Virginia, West Virginia, and Wyoming.

  Class 2 senators consist of the 33 senators whose seats are scheduled for re-election in November 2014 and whose terms end in January 2015 as found in [4].

- Class 3

  States with a Class 3 senator: Alabama, Alaska, Arizona, Arkansas, California, Colorado, Connecticut, Florida, Georgia, Hawaii, Idaho, Illinois, Indiana, Iowa, Kansas, Kentucky, Louisiana, Maryland, Missouri, Nevada, New Hampshire, New York, North Carolina, North Dakota, Ohio, Oklahoma, Oregon, Pennsylvania, South Carolina, South Dakota, Utah, Vermont, Washington, and Wisconsin.

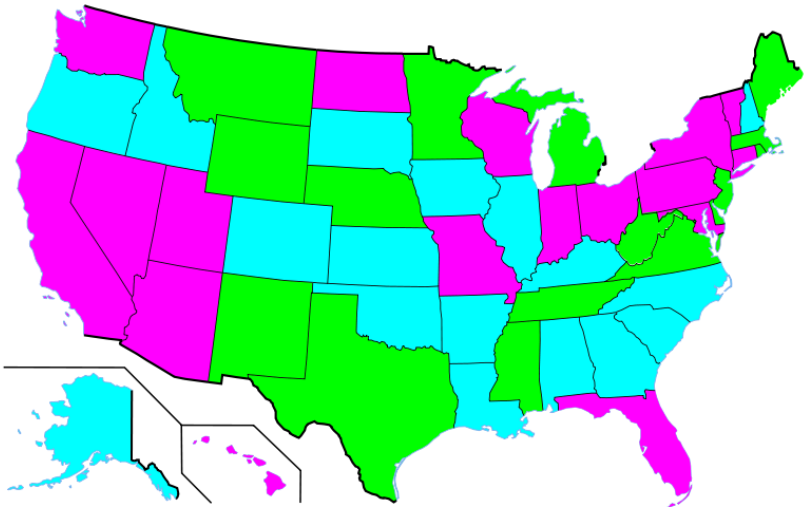
  Class 3 senators consist of the 33 senators whose seats are scheduled for re-election in November 2016, and whose terms end in January 2017 as found in [4].

  Below is the pictorial representation of the state based classes of the senate (Figure 2.1).

2.4 POWERS OF SENATOR OVER HOUSE

Below is the list of powers of the senate when compared to the house.

- Consenting to treaties as a precondition to their ratification.



Map shows the classes in each US State:
Classes 1 and 2
Classes 1 and 3
Classes 2 and 3

Figure 2.1. Senate classes in each state.

- Consenting to or confirming appointments of Cabinet secretaries federal judges, other federal executive officials, military officers, regulatory officials, ambassadors, and other federal uniformed officers as well as trial of federal officials impeached by the House. A bestselling book about senate, years back, was advice and consent.
- The senate is both more deliberative and more prestigious than the House of Representatives, due to its longer terms, smaller size, and statewide constituencies, which historically led to a more collegial and less partisan atmosphere.
- The house has its own authority, for example, spending plans must originate in the house. The nations founding father found government Senate is both a more deliberative and its members are more prestigious than the House of Representatives, due to its longer terms, smaller size, and statewide constituencies, which historically led to a more collegial and less partisan atmosphere.

CHAPTER 3

TECHNOLOGY

The U.S senate application is developed on the technology that supports mobile/tablet based applications development. In the mobile world, the front-runners like iPhone and Android form the major market shareholders. U.S. senator application is targeted for students and teacher of different grades who would like refreshing their memory on U.S senators of different states over different time periods. This application is developed specifically for Android based Mobiles/Tablet. Based on the 3rd quarter result of mobile market in 2012, Android forms 72.2 % of the mobile market, out ranking its competitors. The application is developed using Android SDK and JAVA programming language. Most of the Android native applications are developed using JAVA. Android NDK forms the lower layer of the Architecture and uses C++ for development purpose. The application uses SQLlite for database related support. The layout for each screens is developed using XML scripting language. Below is the consolidated list of technologies used for developing the senator's app for Android devices.

- Tools – Android SDK (*Android debug bridge* (adb))
- Integrated Development Environment (IDE) – Eclipse (*Android virtual devices* (AVD))
- Programming Language - JAVA
- Database - SQLite
- User Interface (UI) – XML

Important point to note - it's possible to develop Android applications without Eclipse and the Android Developer Tools plug-in, but you would need to know your way around the Android SDK. Next subsection will discuss in detail about each of the technologies, its purpose and usage.

3.1 ANDROID OVERVIEW

This section gives an overall idea about android, its meaning, evolution, plugins, platform and its architecture. The section also covers the different set of tools that comes as

part of the android software development kit (SDK) like debugger, libraries and a handset emulator that was an integral part of the development environment.

3.1.1 About Android

Android is a complete operating environment based upon the Linux® V2.6 kernel. Initially, the deployment target for Android was the mobile-phone arena, including smart phones and lower-cost flip-phone devices. However, Android's full range of computing services and rich functional support has the potential to extend beyond the mobile-phone market. Android being an open-source software stack for mobile devices, it offers you a great deal of customization in widgets and over millions of apps. Most of them are free of cost and can be installed on your phone just by clicking on install tab of the respective app in the Google Play Store app. Below is the android logo as found in [5]: See Figure 3.1.

Figure 3.1. Android logo.

3.1.2 Name – ‘Android’

Android has a dictionary meaning of being a human that resembles automation. The true character of its name is it’s the Google created software stack for creating comprehensive Mobile Applications and Software to realize the full potential of one’s Mobile handset and its possibilities. Android is a comprehensive software stack of mobile devices that includes an operating system, middleware and key applications. This software bundle is used in Mobile Technology and is called The Android Software Development Kit (SDK) as found in [6].

3.1.3 Evolution of Android

Andy Rubin founded Android, Inc. in Palo Alto, California in October 2003. Google acquired Android Inc. on August 17, 2005, making it a wholly owned subsidiary of Google. Key employees of Android Inc., including Rubin, Miner and White, stayed at the company after the acquisition. The Android platform is the product of the Open Handset Alliance, a group of organizations collaborating to build a better mobile phone. The group, led by Google, includes mobile operators, device handset manufacturers, component manufacturers, software solution and platform providers, and marketing companies as found [7].

3.1.4 Phone Manufactures that Use Android

Android is an open source platform which can be used by any phone manufacturers on the world. Samsung is the largest manufacturer of android phones and tablets. LG, HTC, Sony, are other top manufacturers of android phones and tablets. Some local manufacturers like Micromax, Karbon, Hawaii, also use android Phones on their portable devices.

3.1.5 Android Version

Android is released in a series of versions. Starting from 1.0 version (where 2.0, 3.0 etc are latest releases). Google names these versions with some food items like ice cream, jellybean, donut etc. that is one of the specialties of android versions. Below is the figure that showcases different versions of android. Jellybean is the most latest version that is missing the below image (Figure 3.2).

The U.S senator Android application supports both mobile and tablet specifications. To aid both devices Android provides support for multiple screen sizes and densities, reflecting the many different screen configurations that a device may have.

To simplify the way that you design your user interfaces for multiple screens, Android divides the range of actual screen sizes and densities into the groups mentioned below:

- The set of four generalized **sizes**: *small*, *normal*, *large*, and *xlarge*. Beginning with Android 3.2 (API level 13), these size groups are deprecated in favor of a new

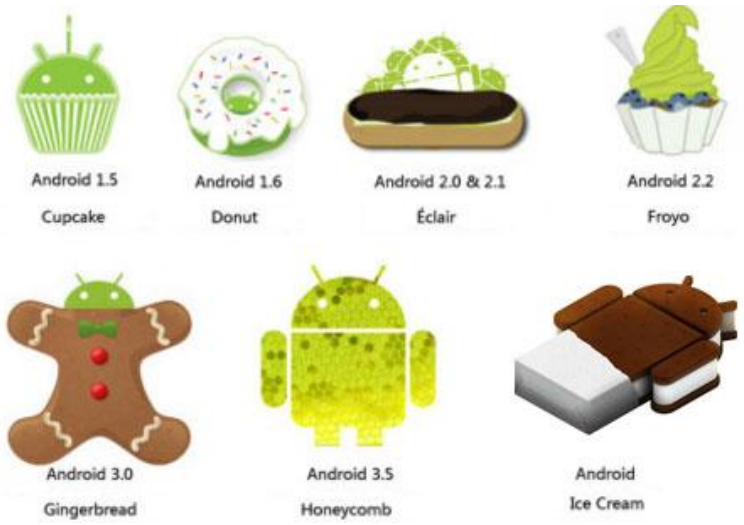


Figure 3.2. Android version.

technique for managing screen sizes based on the available screen width. For the first generation of tablets running Android 3.0, the proper way to declare tablet layouts was to put them in a directory with the `xlarge` configuration qualifier (for example, `res/layout-xlarge/`)

- A set of four generalized **densities**: *ldpi* (low), *mdpi* (medium), *hdpi* (high), and *xhdpi* (extra high)

3.1.6 Android Architecture

Android runs atop a Linux kernel. Android applications are written in the Java programming language, and they run within a virtual machine (VM). It's important to note that the VM is not a JVM as you might expect, but is the Dalvik Virtual Machine, an open source technology. Each Android application runs within an instance of the Dalvik VM, which in turn resides within a Linux-kernel managed process, as shown in Figure 3.3.

3.1.7 Dalvik Virtual Machine

The Android system uses a special virtual machine, i.e. the *Dalvik Virtual Machine* to run Java based applications. Dalvik uses its own byte code format, which is different from Java byte code. Therefore we cannot directly run Java class files on Android; they need to get converted in to the Dalvik byte code format.

3.2 ANDROID SOFTWARE DEVELOPMENT KIT

The *Android Software Development Kit* (SDK) contains the necessary tools to create, compile and package an Android application. The Android SDK also provides an

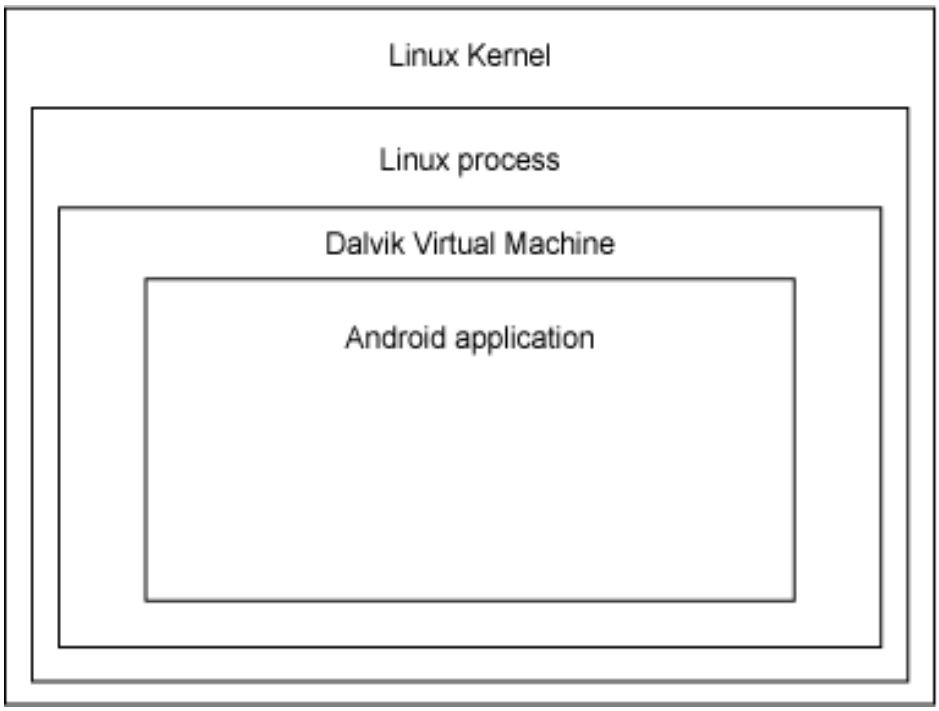


Figure 3.3. Android architecture

Android device emulator, so that Android applications can be tested without a real Android phone. You can create *Android virtual devices* (AVD) via the Android SDK, which runs in this emulator. The Android SDK contains the *Android debug bridge* (adb) tool, which allows connecting to a virtual or real Android device. Most of these tools are command line based.

3.2.1 Overcome of Traditionally Challenges

Historically, two areas where mobile applications have struggled to keep pace with their desktop counterparts are graphics/media, and data storage methods. Android addresses the graphics challenge with built-in support for 2-D and 3-D graphics, including the OpenGL library. The data-storage burden is eased because the Android platform includes the popular open source SQLite database (Figure 3.4).

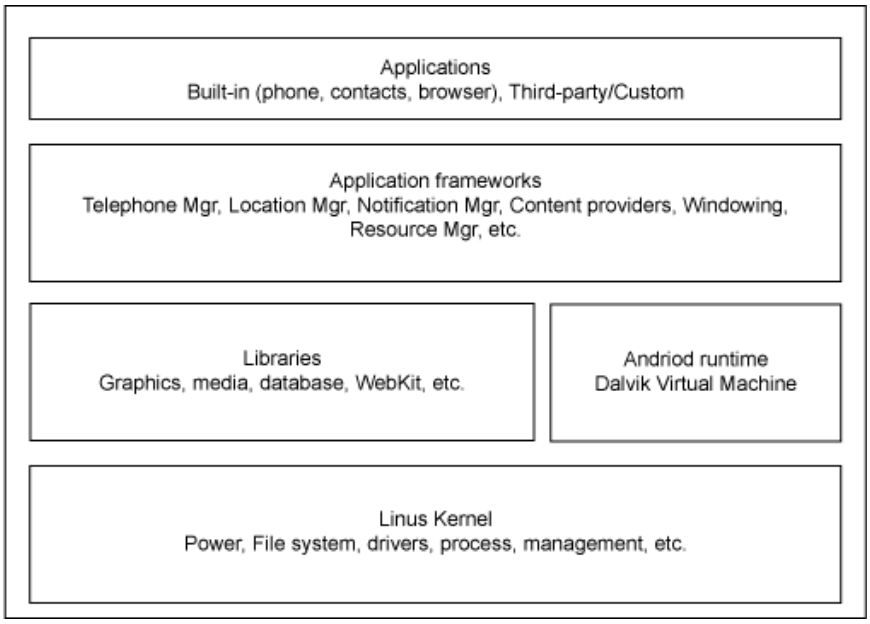

Applications
Built-in (phone, contacts, browser), Third-party/Custom
Application frameworks
Telephone Mgr, Location Mgr, Notification Mgr, Content providers, Windowing, Resource Mgr, etc.
Libraries
Graphics, media, database, WebKit, etc.
Andriod runtime
Dalvik Virtual Machine
Linus Kernel
Power, File system, drivers, process, management, etc.

Figure 3.4. Simplified android software view.

3.2.2 Android Emulator

The Android SDK includes a mobile device emulator — a virtual mobile device that runs on your computer. The emulator lets you develop and test Android applications without using a physical device as found in [8]. The emulator supports a variety of options that you can specify when launching the emulator, to control its appearance or behavior. Figure 3.5 is a sample image of an emulator as found in [9].

3.2.3 Android Debug Bridge

Android Debug Bridge (adb) is a versatile command line tool that lets you communicate with an emulator instance or connected Android-powered device. It is a client-server program that provides three components:

- A client, which runs on your development machine. You can invoke a client from a shell by issuing an adb command. Other Android tools such as the ADT plugin and DDMS also create adb clients.
- A server, which runs as a background process on your development machine. The server manages communication between the client and the adb daemon running on an emulator or device.
- A daemon, which runs as a background process on each emulator or device instance. Above discussed points can be referred from [10].

Figure 3.5. Android emulator.

3.2.4 Android Development Tools (ADT)

Android Development Tools (ADT) is a plugin for the Eclipse IDE that is designed to give a powerful, integrated environment in which we can build Android applications. ADT extends the capabilities of Eclipse to let us quickly set up new Android projects, create an application UI, add packages based on the Android Framework API, debug your applications using the Android SDK tools, and even export signed (or unsigned) `.Apk` files in order to distribute your application. Developing in Eclipse with ADT is highly recommended and is the fastest way to get application development started. With the guided project setup it provides, as well as tools integration, custom XML editors, and debug output pane, ADT gives you an incredible boost in developing Android applications as found in [11].

3.3 ECLIPSE

Eclipse is a multi-language software development environment comprising a base workspace and an extensible plug-in system for customizing the environment. It is written

mostly in Java. It can be used to develop applications in Java, by means of various plug-ins.

3.4 SQLITE OVERVIEW

SQLite is an Open Source Database, which is embedded into Android. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. In addition it requires only a little memory at runtime (approx. 250 KByte). SQLite is available on every Android device. Using an SQLite database in Android does not require any database setup or administration. SQLite is fast and has a small footprint, making it perfect for Android devices. Android applications can choose to store private application data in a SQLite database. SQLite is basically a stripped-down SQL database engine for embedded devices as found in [12].

CHAPTER 4

THE REQUIREMENTS

This Application mainly focuses on the development of an Android native app that aids political science students and teachers of different school grades to quickly browse through senator details. This application covers list of senators from different states over a century for states that had a long senate history in U.S. The Application is developed on the latest technology that makes it more accessible to Mobile enthusiasts and the younger generation, to develop their knowledge in the area of political science. Most of the senate related information is contained in this application, which makes it more desirable and user friendly for different age groups of people. Due to the applications being self-contained with the use of a lightweight database, the use of the Internet can be avoided, which saves money and is hassle free. This gives more freedom and makes it portable to different devices like android tablets, kindle etc. Most of the requirements of this thesis application came from Dr. Carl Eckberg who is a graduate advisor of the Computer Science department in San Diego State University and under his guidance this application was developed.

I have categorized requirements into the following three categories:

1. Data requirements
2. Platform requirements
3. Functional requirements

4.1 DATA REQUIREMENTS

Data requirements mean the data, which needs to be represented. The senator related information is displayed in the form of an interactive map based application which can be downloaded from Professor Eckberg's website and run on specific devices as discussed in the technology and the requirements chapter. When the application is opened on the users device a whole U.S map is displayed. The user is given an option to tap on any part of the U.S map. The map is divided into four regions and each region is divided into two or three divisions. The four regions are used by the United States Census Bureau. Below is a List of regions and the divisions that belong to these regions as found in [13]:

- Region 1 (Northeast)
  - Division 1 (New England) –Maine, New Hampshire, Vermont, Massachusetts, Rhode Island, Connecticut.
  - Division 2 (Mid-Atlantic) – New York, Pennsylvania, New Jersey.
- Region 2 (Midwest) (Prior to June 1984, the Midwest Region was designated as the North Central Region.
  - Division 3 (East North Central) – Wisconsin, Michigan, Illinois, Indiana, Ohio.
  - Division 4 (West North Central) – Missouri, North Dakota, South Dakota, Nebraska, Kansas, Minnesota, Iowa.
- Region 3 (South)
  - Division 5 (South Atlantic) - Delaware, Maryland, District of Columbia, Virginia, West Virginia, North Carolina, South Carolina, Georgia, Florida.
  - Division 6 (East South Central) - Kentucky, Tennessee, Mississippi, Alabama.
  - Division 7 (West South Central) - Oklahoma, Texas, Arkansas, Louisiana.
- Region 4 (West)
  - Division 8 (Mountain) – (Mountain) – Idaho, Montana, Wyoming, Nevada, Utah, Colorado, Arizona, New Mexico.
  - Division 9 (Pacific) – Alaska, Washington, Oregon, California, Hawaii.

4.2 PLATFORM REQUIREMENTS

In platform requirements there is specification of programming language and other technologies used to develop this application. JAVA is used as the programming language for development of this application. Android SDK, with other related plugins that are mentioned in the technology chapter, are used as the development environment. The testing of this application is done in Android Nexus Tablet with Jellybean Version and Samsung Galaxy S3 with Ice cream sandwich on phones. The operating system used for Development was Mac OS (an Apple Product).

4.3 FUNCTIONAL REQUIREMENTS

The functional requirements of this application refer to the specification of the functions the application should perform, and other features the application should possess.

Dr. Carl Eckberg gave all the basic functional requirements and I organized them with some additional features. The list of functional requirements is given below:

- Once the application is downloaded on the device, the user should be able to click on the Senator Launcher Icon to launch the application.
- When the user starts the application the user must be able to see the whole U.S map with the current party based strength for the entire country.
- Tapping on the U.S map must be recognized as one of the four regions namely West, South, Midwest, Northeast. All the four highlighted regions must be color-coded.
- The entire list of current senators must be displayed as Quick Facts on the right hand pane when the user taps on the US map.
- Tapping and holding a specific region must display a division maps with the states that belong to that division of the region.
- Tapping on a particular state must display the list of senators of that state.
- When the user taps on the 'Take a Quiz' button, a quiz screen must be displayed.
- When the user takes the quiz based on the options selected in the quiz, a message must be displayed to indicate if the answer chosen was correct/ wrong.
- When the user taps on the 'Tutorial' button, a tutorial screen about the senate and his/ her responsibilities must be displayed.
- When the user taps on the 'Read me' button, a instruction screen on how to use the application and its features must be displayed.
- When the user taps on the 'Advanced Search' button, an advanced search screen must be displayed with 'All' as default option.
- In the 'Advanced Search' screen, user must be able to filter based on the state.
- In the 'Advanced Search' screen, user must be able to filter based on the senate class.
- User must be able to arrange the senator name in both ascending and descending order.
- User must be able to use the application both on phone and tablet on both Ice-cream sandwich and Jellybean.

CHAPTER 5

ANALYSIS

I had analyzed all the requirements needed for the development of this application and came up this list of steps.

1. Understand the U.S congress and its constitutional structure. Focus on the Senator part of the congress. Understand the core responsibilities and functioning of the U.S senate.
2. Make a plan for organizing the senators details based on either the geographical regions of the U.S or Geo-political division or division based on the party strength of each state.
3. Then based on the decision, collect the list of senators from each state from 1850 to present. In our case collect the list based on geographical and geo-political division of the U.S.
4. Once the data is collected from the mentioned sites in chapter 6 save them as text files, in the order in which you would like them to be displayed in the application. Then convert the text files into the xml format using convertor software. The xml files are easier understood by the Android application.
5. Keep in mind the final application is displayed in mobile/ tablet devices, so the road map of the flow of application must cater to a smaller real estate than regular desktop application.
6. Make the necessary images for each state based on the regional classification using GIMP software. It would be easier to recognize if the grouped state are color-coded.
7. Set up the development environment based on the user manual chapter (see Appendix A).
8. The Application is supported only on the devices that run on Ice-cream sandwich and above in both phone and tablet.
9. Have a basic plan for the flow of the application with the placement of different widgets. Make sure user interface is intuitive and provide an instruction guide, to help users maneuver the application and explore all the features.
10. Develop the main screen of the application so that there is a starting point for the application.
11. Develop the layout of a particular screen and the corresponding background connection using java to display the screen.
12. Add the xml data collected to the development environment to populate the collected information regions wise.

13. When the user taps on to a specific region of the U.S map, display the states that belong to that region. Further tapping on to a particular state in the group display the senator related details. The step-by-step flow in the application gives user a path to follow and help them use the application.
14. Provide a user quiz feature that helps them recollect the information in the application.

CHAPTER 6

DATA COLLECTION AND PREPARATION OF DATA

This thesis application needed a large amount of data to be collected for different state senators over a century. Please note that some of the states like Arizona the data provided is from the 20th century.

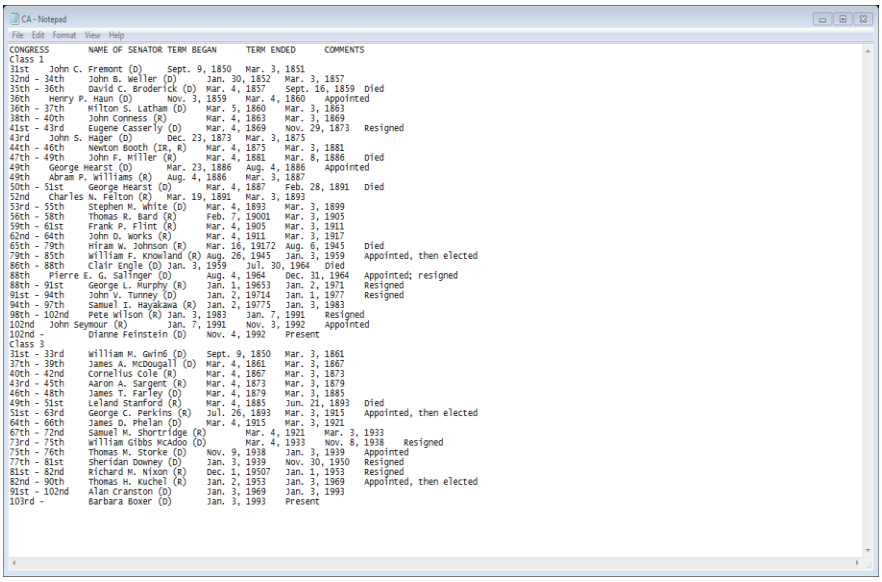
The application need data based on each state in the U.S. Professor Carl Eckberg (Department of computer Science) guided me in organizing the information which could be displayed in a more user friendly way. Below are the links of different states from the senator.gov site [14], an official site for U.S senators. In addition, I have also added more relevant columns for each state based senator list (Figure 6.1).

Initially, when the application starts a U.S map is displayed. Based on the user's selection regional division map is displayed. Then the state selection displays the information provided in each of the links above. The information provided in the links is saved as a flat file. The screen shot of the text file for the state of California is in Figure 6.2. Each of the text files has both the classes of the senator of a particular state.

Using the text file for senators of each state, the xml-formatted files were generated using the conversion software. Below is the sample screen shot for senators of Washington State, Montana, Idaho and Wyoming (Figures 6.3-6.6).

State based U.S Senator List- Links in Ascending order		
S.No	**State**	**URL**
1	Alabama	http://www.senate.gov/states/AL/senators.htm
2	Alaska	http://www.senate.gov/states/AK/senators.htm
3	Arizona	http://www.senate.gov/states/AZ/senators.htm
4	Arkansas	http://www.senate.gov/states/AR/senators.htm
5	California	http://www.senate.gov/states/CA/senators.htm
6	Colorado	http://www.senate.gov/states/CO/senators.htm
7	Connecticut	http://www.senate.gov/states/CT/senators.htm
8	Delaware	http://www.senate.gov/states/DE/senators.htm
9	Florida	http://www.senate.gov/states/FL/senators.htm
10	Georgia	http://www.senate.gov/states/GA/senators.htm
11	Hawaii	http://www.senate.gov/states/HI/senators.htm
12	Idaho	http://www.senate.gov/states/ID/senators.htm
13	Illinois	http://www.senate.gov/states/IL/senators.htm
14	Indiana	http://www.senate.gov/states/IN/senators.htm
15	Iowa	http://www.senate.gov/states/IA/senators.htm
16	Kansas	http://www.senate.gov/states/KS/senators.htm
17	Kentucky	http://www.senate.gov/states/KY/senators.htm
18	Louisiana	http://www.senate.gov/states/LA/senators.htm
19	Maine	http://www.senate.gov/states/ME/senators.htm
20	Maryland	http://www.senate.gov/states/MD/senators.htm
21	Massachusetts	http://www.senate.gov/states/MA/senators.htm
22	Michigan	http://www.senate.gov/states/MI/senators.htm
23	Minnesota	http://www.senate.gov/states/MN/senators.htm
24	Mississippi	http://www.senate.gov/states/MS/senators.htm
25	Missouri	http://www.senate.gov/states/MO/senators.htm
26	Montana	http://www.senate.gov/states/MT/senators.htm
27	Nebraska	http://www.senate.gov/states/NE/senators.htm
28	Nevada	http://www.senate.gov/states/NV/senators.htm

Figure 6.1. State based senator URL's.

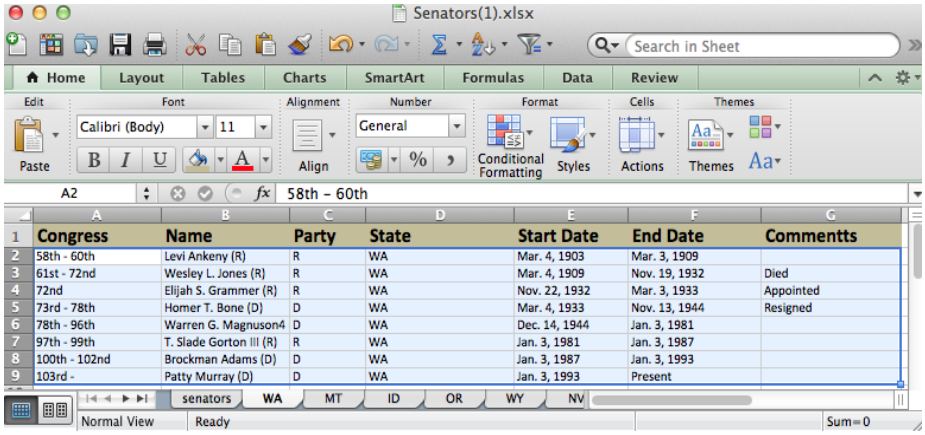


CA - Notepad

File Edit Format View Help

```
CONGRESS        NAME OF SENATOR TERM BEGAN      TERM ENDED      COMMENTS
Class 1
31st    John C. Fremont (D)     Sept. 9, 1850   Mar. 3, 1851
32nd - 34th     John B. Weller (D)      Jan. 30, 1852   Mar. 3, 1857
35th - 36th     David C. Broderick (D)  Mar. 4, 1857    Sept. 16, 1859  Died
36th    Henry P. Haun (D)       Nov. 3, 1859    Mar. 4, 1860    Appointed
36th - 37th     Milton S. Latham (D)    Mar. 5, 1860    Mar. 3, 1863
38th - 40th     John Conness (R)        Mar. 4, 1863    Mar. 3, 1869
41st - 43rd     Eugene Casserly (D)     Mar. 4, 1869    Nov. 29, 1873   Resigned
43rd    John S. Hager (D)       Dec. 23, 1873   Mar. 3, 1875
44th - 46th     Newton Booth (IR, R)    Mar. 4, 1875    Mar. 3, 1881
47th - 49th     John F. Miller (R)      Mar. 4, 1881    Mar. 8, 1886    Died
49th    George Hearst (D)       Mar. 23, 1886   Aug. 4, 1886    Appointed
49th    Abram P. Williams (R)   Aug. 4, 1886    Mar. 3, 1887
50th - 51st     George Hearst (D)       Mar. 4, 1887    Feb. 28, 1891   Died
52nd    Charles N. Felton (R)   Mar. 19, 1891   Mar. 3, 1893
53rd - 55th     Stephen M. White (D)    Mar. 4, 1893    Mar. 3, 1899
56th - 58th     Thomas R. Bard (R)      Feb. 7, 19001   Mar. 3, 1905
59th - 61st     Frank P. Flint (R)      Mar. 4, 1905    Mar. 3, 1911
62nd - 64th     John D. Works (R)       Mar. 4, 1911    Mar. 3, 1917
65th - 79th     Hiram W. Johnson (R)    Mar. 16, 19172  Aug. 6, 1945    Died
79th - 85th     William F. Knowland (R) Aug. 26, 1945   Jan. 3, 1959    Appointed, then elected
86th - 88th     Clair Engle (D) Jan. 3, 1959    Jul. 30, 1964   Died
88th    Pierre E. G. Salinger (D)       Aug. 4, 1964    Dec. 31, 1964   Appointed; resigned
88th - 91st     George L. Murphy (R)    Jan. 1, 19653   Jan. 2, 1971    Resigned
91st - 94th     John V. Tunney (D)      Jan. 2, 19714   Jan. 1, 1977    Resigned
94th - 97th     Samuel I. Hayakawa (R)  Jan. 2, 19775   Jan. 3, 1983
98th - 102nd    Pete Wilson (R) Jan. 3, 1983    Jan. 7, 1991    Resigned
102nd   John Seymour (R)         Jan. 7, 1991    Nov. 3, 1992    Appointed
102nd -         Dianne Feinstein (D)    Nov. 4, 1992    Present
Class 3
31st - 33rd     William M. Gwin6 (D)    Sept. 9, 1850   Mar. 3, 1861
37th - 39th     James A. McDougall (D)  Mar. 4, 1861    Mar. 3, 1867
40th - 42nd     Cornelius Cole (R)      Mar. 4, 1867    Mar. 3, 1873
43rd - 45th     Aaron A. Sargent (R)    Mar. 4, 1873    Mar. 3, 1879
46th - 48th     James T. Farley (D)     Mar. 4, 1879    Mar. 3, 1885
49th - 51st     Leland Stanford (R)     Mar. 4, 1885    Jun. 21, 1893   Died
51st - 63rd     George C. Perkins (R)   Jul. 26, 1893   Mar. 3, 1915    Appointed, then elected
64th - 66th     James D. Phelan (D)     Mar. 4, 1915    Mar. 3, 1921
67th - 72nd     Samuel M. Shortridge (R)        Mar. 4, 1921    Mar. 3, 1933
73rd - 75th     William Gibbs McAdoo (D)        Mar. 4, 1933    Nov. 8, 1938    Resigned
75th - 76th     Thomas M. Storke (D)    Nov. 9, 1938    Jan. 3, 1939    Appointed
77th - 81st     Sheridan Downey (D)     Jan. 3, 1939    Nov. 30, 1950   Resigned
81st - 82nd     Richard M. Nixon (R)    Dec. 1, 19507   Jan. 1, 1953    Resigned
82nd - 90th     Thomas H. Kuchel (R)    Jan. 2, 1953    Jan. 3, 1969    Appointed, then elected
91st - 102nd    Alan Cranston (D)       Jan. 3, 1969    Jan. 3, 1993
103rd -         Barbara Boxer (D)       Jan. 3, 1993    Present
```

Figure 6.2. California state senator text file.



Senators(1).xlsx

Search in Sheet

Home | Layout | Tables | Charts | SmartArt | Formulas | Data | Review

Edit | Font | Alignment | Number | Format | Cells | Themes

Paste | Calibri (Body) | 11 | Align | General | Conditional Formatting | Styles | Actions | Themes

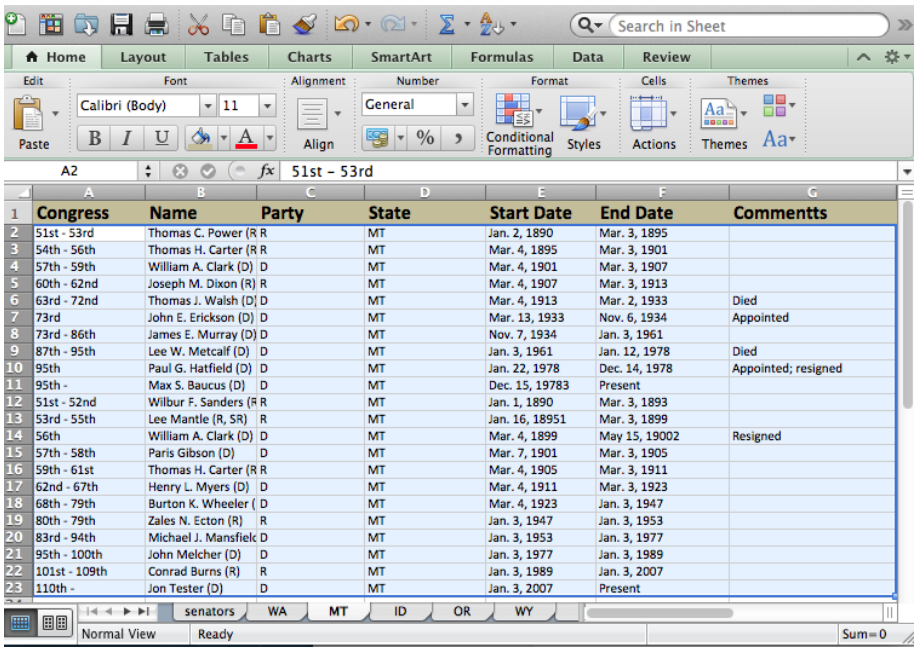
A2 | 58th – 60th

	A	B	C	D	E	F	G
1	Congress	Name	Party	State	Start Date	End Date	Commentts
2	58th - 60th	Levi Ankeny (R)	R	WA	Mar. 4, 1903	Mar. 3, 1909	
3	61st - 72nd	Wesley L. Jones (R)	R	WA	Mar. 4, 1909	Nov. 19, 1932	Died
4	72nd	Elijah S. Grammer (R)	R	WA	Nov. 22, 1932	Mar. 3, 1933	Appointed
5	73rd - 78th	Homer T. Bone (D)	D	WA	Mar. 4, 1933	Nov. 13, 1944	Resigned
6	78th - 96th	Warren G. Magnuson4	D	WA	Dec. 14, 1944	Jan. 3, 1981	
7	97th - 99th	T. Slade Gorton III (R)	R	WA	Jan. 3, 1981	Jan. 3, 1987	
8	100th - 102nd	Brockman Adams (D)	D	WA	Jan. 3, 1987	Jan. 3, 1993	
9	103rd -	Patty Murray (D)	D	WA	Jan. 3, 1993	Present	

senators | WA | MT | ID | OR | WY | NV

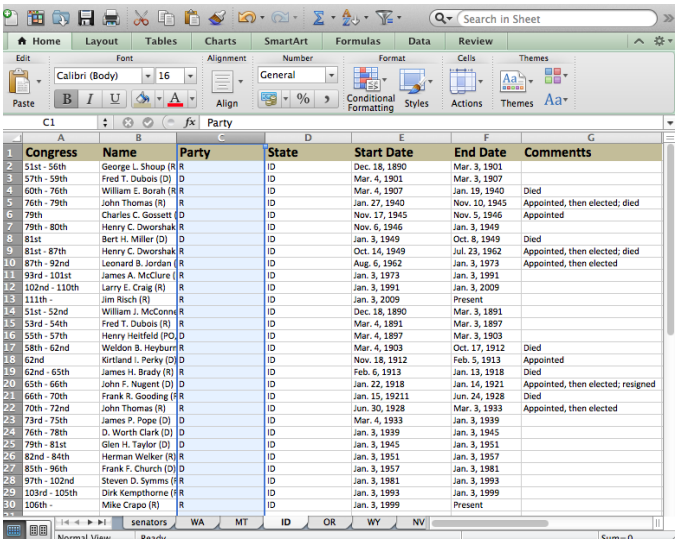
Normal View | Ready | Sum=0

Figure 6.3. Senator information for state of Washington.



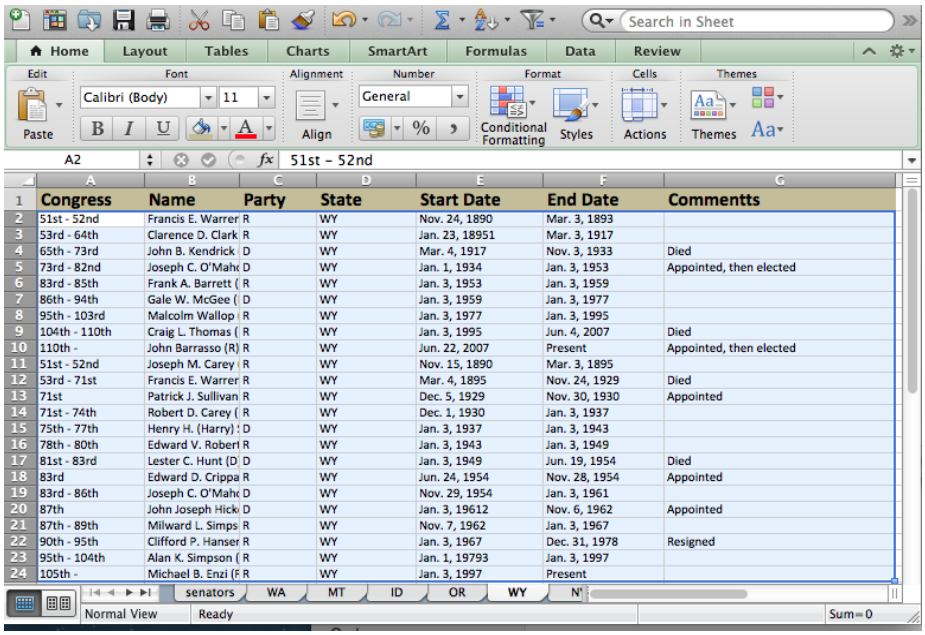
Congress	Name	Party	State	Start Date	End Date	Commentts
51st - 53rd	Thomas C. Power (R	R	MT	Jan. 2, 1890	Mar. 3, 1895	
54th - 56th	Thomas H. Carter (R	R	MT	Mar. 4, 1895	Mar. 3, 1901	
57th - 59th	William A. Clark (D)	D	MT	Mar. 4, 1901	Mar. 3, 1907	
60th - 62nd	Joseph M. Dixon (R)	R	MT	Mar. 4, 1907	Mar. 3, 1913	
63rd - 72nd	Thomas J. Walsh (D[	D	MT	Mar. 4, 1913	Mar. 2, 1933	Died
73rd	John E. Erickson (D)	D	MT	Mar. 13, 1933	Nov. 6, 1934	Appointed
73rd - 86th	James E. Murray (D)	D	MT	Nov. 7, 1934	Jan. 3, 1961	
87th - 95th	Lee W. Metcalf (D)	D	MT	Jan. 3, 1961	Jan. 12, 1978	Died
95th	Paul G. Hatfield (D)	D	MT	Jan. 22, 1978	Dec. 14, 1978	Appointed; resigned
95th -	Max S. Baucus (D)	D	MT	Dec. 15, 19783	Present	
51st - 52nd	Wilbur F. Sanders (R	R	MT	Jan. 1, 1890	Mar. 3, 1893	
53rd - 55th	Lee Mantle (R, SR)	R	MT	Jan. 16, 18951	Mar. 3, 1899	
56th	William A. Clark (D)	D	MT	Mar. 4, 1899	May 15, 19002	Resigned
57th - 58th	Paris Gibson (D)	D	MT	Mar. 7, 1901	Mar. 3, 1905	
59th - 61st	Thomas H. Carter (R	R	MT	Mar. 4, 1905	Mar. 3, 1911	
62nd - 67th	Henry L. Myers (D)	D	MT	Mar. 4, 1911	Mar. 3, 1923	
68th - 79th	Burton K. Wheeler (	D	MT	Mar. 4, 1923	Jan. 3, 1947	
80th - 79th	Zales N. Ecton (R)	R	MT	Jan. 3, 1947	Jan. 3, 1953	
83rd - 94th	Michael J. Mansfield	D	MT	Jan. 3, 1953	Jan. 3, 1977	
95th - 100th	John Melcher (D)	D	MT	Jan. 3, 1977	Jan. 3, 1989	
101st - 109th	Conrad Burns (R)	R	MT	Jan. 3, 1989	Jan. 3, 2007	
110th -	Jon Tester (D)	D	MT	Jan. 3, 2007	Present	

Figure 6.4 Senator information for state of Montana.



Congress	Name	Party	State	Start Date	End Date	Commentts
51st - 56th	George L. Shoup (R	R	ID	Dec. 18, 1890	Mar. 3, 1901	
57th - 59th	Fred T. Dubois (D)	D	ID	Mar. 4, 1901	Mar. 3, 1907	
60th - 76th	William E. Borah (R	R	ID	Mar. 4, 1907	Jan. 19, 1940	Died
76th - 79th	John Thomas (R)	R	ID	Jan. 27, 1940	Nov. 10, 1945	Appointed, then elected; died
79th	Charles C. Gossett (	D	ID	Nov. 17, 1945	Nov. 5, 1946	Appointed
79th - 80th	Henry C. Dworshak	R	ID	Nov. 6, 1946	Jan. 3, 1949	
81st	Bert H. Miller (D)	D	ID	Jan. 3, 1949	Oct. 8, 1949	Died
81st - 87th	Henry C. Dworshak	R	ID	Oct. 14, 1949	Jul. 23, 1962	Appointed, then elected; died
87th - 92nd	Leonard B. Jordan (	R	ID	Aug. 6, 1962	Jan. 3, 1973	Appointed, then elected
93rd - 101st	James A. McClure (	R	ID	Jan. 3, 1973	Jan. 3, 1991	
102nd - 110th	Larry E. Craig (R)	R	ID	Jan. 3, 1991	Jan. 3, 2009	
111th -	Jim Risch (R)	R	ID	Jan. 3, 2009	Present	
51st - 52nd	William J. McConne	R	ID	Dec. 18, 1890	Mar. 3, 1891	
53rd - 54th	Fred T. Dubois (R)	R	ID	Mar. 4, 1891	Mar. 3, 1897	
55th - 57th	Henry Heitfeld (PO,	D	ID	Mar. 4, 1897	Mar. 3, 1903	
58th - 62nd	Weldon B. Heyburn	R	ID	Mar. 4, 1903	Oct. 17, 1912	Died
62nd	Kirtland I. Perky (D)	D	ID	Nov. 18, 1912	Feb. 5, 1913	Appointed
62nd - 65th	James H. Brady (R)	R	ID	Feb. 6, 1913	Jan. 13, 1918	Died
65th - 66th	John F. Nugent (D)	D	ID	Jan. 22, 1918	Jan. 14, 1921	Appointed, then elected; resigned
66th - 70th	Frank R. Gooding (R	R	ID	Jan. 15, 19211	Jun. 24, 1928	Died
70th - 72nd	John Thomas (R)	R	ID	Jun. 30, 1928	Mar. 3, 1933	Appointed, then elected
73rd - 75th	James P. Pope (D)	D	ID	Mar. 4, 1933	Jan. 3, 1939	
76th - 78th	D. Worth Clark (D)	D	ID	Jan. 3, 1939	Jan. 3, 1945	
79th - 81st	Glen H. Taylor (D)	D	ID	Jan. 3, 1945	Jan. 3, 1951	
82nd - 84th	Herman Welker (R)	R	ID	Jan. 3, 1951	Jan. 3, 1957	
85th - 96th	Frank F. Church (D)	D	ID	Jan. 3, 1957	Jan. 3, 1981	
97th - 102nd	Steven D. Symms (R	R	ID	Jan. 3, 1981	Jan. 3, 1993	
103rd - 105th	Dirk Kempthorne (R	R	ID	Jan. 3, 1993	Jan. 3, 1999	
106th -	Mike Crapo (R)	R	ID	Jan. 3, 1999	Present	

Figure 6.5. Senator information for state of Idaho.



	A	B	C	D	E	F	G
1	Congress	Name	Party	State	Start Date	End Date	Commentts
2	51st - 52nd	Francis E. Warrer	R	WY	Nov. 24, 1890	Mar. 3, 1893	
3	53rd - 64th	Clarence D. Clark	R	WY	Jan. 23, 18951	Mar. 3, 1917	
4	65th - 73rd	John B. Kendrick	D	WY	Mar. 4, 1917	Nov. 3, 1933	Died
5	73rd - 82nd	Joseph C. O'Mahc	D	WY	Jan. 1, 1934	Jan. 3, 1953	Appointed, then elected
6	83rd - 85th	Frank A. Barrett (	R	WY	Jan. 3, 1953	Jan. 3, 1959	
7	86th - 94th	Gale W. McGee (	D	WY	Jan. 3, 1959	Jan. 3, 1977	
8	95th - 103rd	Malcolm Wallop (	R	WY	Jan. 3, 1977	Jan. 3, 1995	
9	104th - 110th	Craig L. Thomas (	R	WY	Jan. 3, 1995	Jun. 4, 2007	Died
10	110th -	John Barrasso (R)	R	WY	Jun. 22, 2007	Present	Appointed, then elected
11	51st - 52nd	Joseph M. Carey (	R	WY	Nov. 15, 1890	Mar. 3, 1895	
12	53rd - 71st	Francis E. Warrer	R	WY	Mar. 4, 1895	Nov. 24, 1929	Died
13	71st	Patrick J. Sullivan	R	WY	Dec. 5, 1929	Nov. 30, 1930	Appointed
14	71st - 74th	Robert D. Carey (	R	WY	Dec. 1, 1930	Jan. 3, 1937	
15	75th - 77th	Henry H. (Harry) (	D	WY	Jan. 3, 1937	Jan. 3, 1943	
16	78th - 80th	Edward V. Robert	R	WY	Jan. 3, 1943	Jan. 3, 1949	
17	81st - 83rd	Lester C. Hunt (D	D	WY	Jan. 3, 1949	Jun. 19, 1954	Died
18	83rd	Edward D. Crippa	R	WY	Jun. 24, 1954	Nov. 28, 1954	Appointed
19	83rd - 86th	Joseph C. O'Mahc	D	WY	Nov. 29, 1954	Jan. 3, 1961	
20	87th	John Joseph Hick	D	WY	Jan. 3, 19612	Nov. 6, 1962	Appointed
21	87th - 89th	Milward L. Simps	R	WY	Nov. 7, 1962	Jan. 3, 1967	
22	90th - 95th	Clifford P. Hanser	R	WY	Jan. 3, 1967	Dec. 31, 1978	Resigned
23	95th - 104th	Alan K. Simpson (	R	WY	Jan. 1, 19793	Jan. 3, 1997	
24	105th -	Michael B. Enzi (F	R	WY	Jan. 3, 1997	Present	

Figure 6.6. Senator information for state of Wyoming.

CHAPTER 7

APPLICATION DEVELOPMENT WITH FEATURES

7.1 IMPLEMENTAION

This application is developed using Eclipse IDE and Android SDK plugins. All the layouts are written using the XML files, connections to inflate the layout and perform the action corresponding to the tapping is written using the JAVA programming language. The data that is populated during each tap is retrieved from the internal database. This database is built into the application using SQLite Database. The screen shot below showcases the Structure of the Project in Eclipse. Each of the functionalities in the application is grouped as Packages (Figure 7.1).

7.2 MAIN APPLICATION SCREEN FLOW - SCREENS

This session discusses about the flow of the application for each of the screen designed as part of the layouts using XML and the corresponding connections to make it interactive. Each sub section will display the screen, its description, How to get to that screen and the next action that needs to be performed on that screen to show the flow of the application. The code corresponding to the development of the screen both XML and JAVA will be provided in each of the below sections.

7.2.1 Main Screen

Description: The main screen is displayed at the start of the application. This screen displays the U.S map with current senators count based on the political parties. In the right pane it displays the current list of all the senators in U.S along with the party they represent.

How to get to the current screen: after downloading the application. Tap on the Senator Icon to open the application.

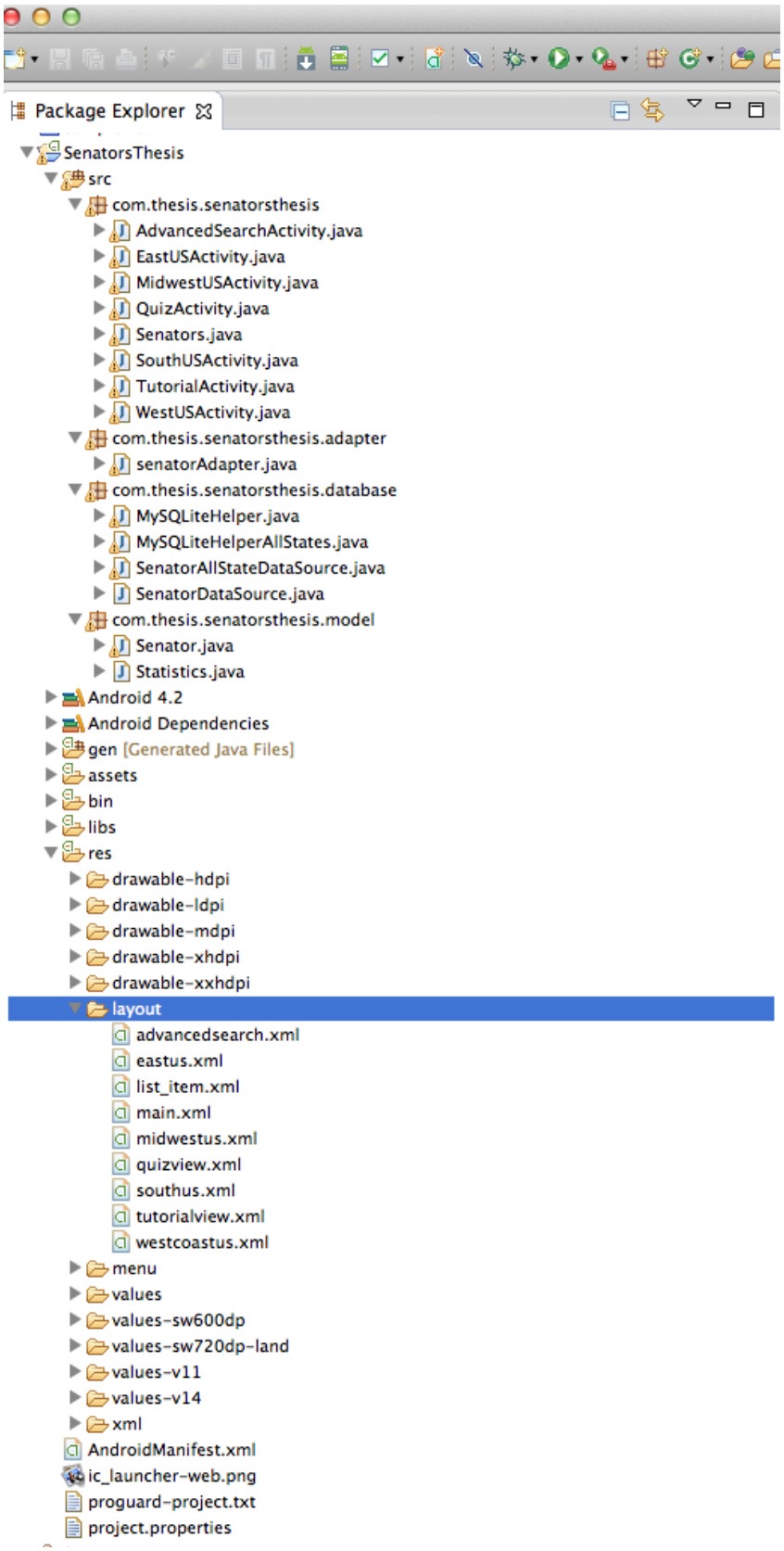


Figure 7.1. Senator application project structure in eclipse.

Next action to be performed – users can choose to tap on any region on the U.S map and hold to select the desired region. The precise structure of the main US map layout is shown in Figure 7.2.

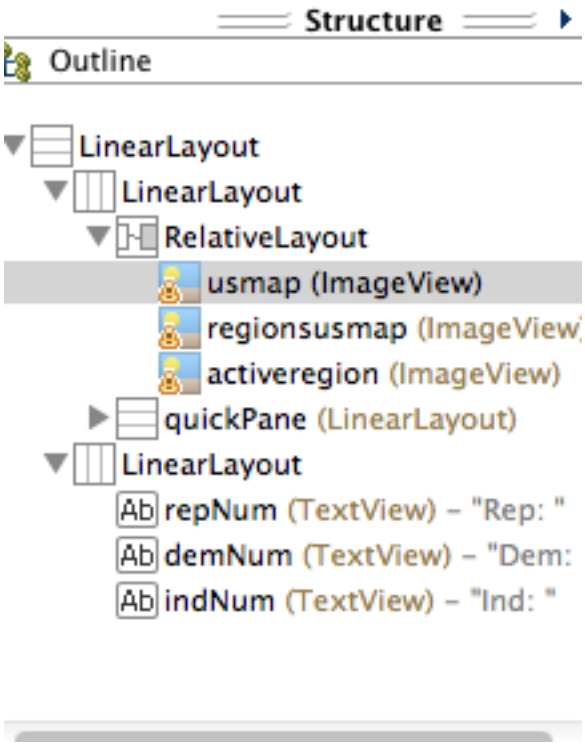


Figure 7.2. Main US map layout structure.

The main screen layout uses a combination of nested linear layout with relative layout to get the appropriate display. There are various widget views that are used to display an image and different information text. The complete code snippet of the main.xml file is given in Appendix B.

Some of the core logic for inflating the XML layout and other connection related to the initial screen is discussed below. The corresponding JAVA code snippets are displayed.

The code snippet of the logic that inflates the main.xml is given below.

```
//senator.java – Refer this class file  for complete code
private Regions selectedRegion;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
```

```
        initialize();

        }
```

Once the layout is loaded, the handle for all the views widgets is obtained. Handle for different widget is performed in initialize method.

```
    //senator.java – Refer this class file for complete code
usMap = (ImageView) findViewById(R.id.usmap);
        usMapRegions = (ImageView) findViewById(R.id.regionsusmap);
        activeRegion = (ImageView) findViewById(R.id.activeregion);

        usMapDrawable = usMap.getBackground();
        usMapBitmap = ((BitmapDrawable)usMap.getBackground()).getBitmap();

        quickPane = (LinearLayout) findViewById(R.id.quickPane);
        cancelButton = (Button) findViewById(R.id.cancelButton);
        moreButton = (Button) findViewById(R.id.moreButton);
        quizButton = (Button) findViewById(R.id.quizButton);
        listView = (ListView) findViewById(R.id.quickList);
        RepView = (TextView) findViewById(R.id.repNum);
        DemView = (TextView) findViewById(R.id.demNum);
            IndView = (TextView) findViewById(R.id.indNum);
```

After getting the handle of different widgets the on click listeners are set so that the appropriate action are lined to the widgets.

```
    //senator.java – Refer this class file for complete code
//          Set on click listners
        cancelButton.setOnClickListener(this);
        moreButton.setOnClickListener(this);
        quizButton.setOnClickListener(this);
        usMap.setOnTouchListener(this);
        usMap.setOnLongClickListener(this);
```

The database is initialized by clearing the statistic and opening the database for use.

```
    //senator.java – Refer this class file for complete code
Statistics.clear();
        datasource = new SenatorDataSource(this);
        datasource.open();
```

The code below is the method that is used to display the quick fact details based on the region tapped.

//senator.java – Refer this class file for complete code

```
protected void SetUpQuickPane(Regions regions, int visible) {
		quickPane.setVisibility(visible);
		activeRegion.setVisibility(visible);
		if(visible==View.VISIBLE)
		{
			ArrayAdapter<Senator> adapter = (ArrayAdapter<Senator>)
listView.getAdapter();
			switch (regions) {
			case WEST:

	quickPane.setBackgroundColor(getResources().getColor(R.color.LightBlue));

	activeRegion.setBackgroundResource(R.drawable.regions_west_colormap);
				break;
			case MIDWEST:

	quickPane.setBackgroundColor(getResources().getColor(R.color.LightYellow));

	activeRegion.setBackgroundResource(R.drawable.regions_midwest_colormap);
				break;
			case SOUTH:

	quickPane.setBackgroundColor(getResources().getColor(R.color.LightPink));

	activeRegion.setBackgroundResource(R.drawable.regions_south_colormap);
				break;
			case EAST:

	quickPane.setBackgroundColor(getResources().getColor(R.color.LightCyan));

	activeRegion.setBackgroundResource(R.drawable.regions_east_colormap);
				break;
			default:
				break;
			}
			adapter.notifyDataSetChanged();
		}
		else
		{
			clear(numTaps);
		}
	}
```

7.2.2 Region Selection Screen

Description: This screen displays the region-based selection. Based on the tap, the application recognizes the region and highlights that region as West, Northeast, Midwest and South. Internally below the visible U.S map is a color-coordinated map based on the region, which is invisible to the user. When the user taps on to a particular region, the hidden map below recognizes the color of the x and y coordinate tapped and checks to see which region the tapped area belongs to and then displays an image of that map with the regions named and color coded. Note – Tapping on it will display the highlighted region and tapping it once again will display the main screen. Tapping and holding will take the user to the next screen.

How to get to the current screen: Tapping and holding on any region on the main screen will display the current screen. Single tap will just highlight the region and tapping the highlighted region will display the main screen.

Next Action to be performed – Tapping on to the regions state (divisions available) will display the senator related information of that state.

Xml code for UI – Refer Appendix B

JAVA code for Logical flow –

//EastUSActivity.java

Below code inflates the eastus.xml

```
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.eastus);

        initialize();
        }
```

Once the xml is inflated handle for each widget view is obtained using the below code. This is code in the initialize method.

```
private void initialize() {
        westUS = (ImageView) findViewById(R.id.westusmap);
        westUSDrawable = westUS.getBackground();
        westUSBitmap = ((BitmapDrawable)westUS.getBackground()).getBitmap();
        listView = (ListView) findViewById(R.id.quickList);
        quickPane = (LinearLayout) findViewById(R.id.quickPane);
        cancelButton = (Button) findViewById(R.id.cancelButton);
```

```
moreButton = (Button) findViewById(R.id.moreButton);
regionName = (TextView) findViewById(R.id.regionName);


westUSStates = (ImageView) findViewById(R.id.westusstates);
selectedState = (ImageView) findViewById(R.id.selectedstate);

westUS.setOnTouchListener(this);
cancelButton.setOnClickListener(this);
moreButton.setOnClickListener(this);
moreButton.setText("Advanced Search");


datasource = new SenatorAllStateDataSource(this);
datasource.open();
```

On tapping a particular region, a set of action is performed and the code below indicates the same. The x and y coordinates are recognized and the appropriate action is performed.

```
@Override
        public boolean onTouch(View arg0, MotionEvent arg1) {
                ArrayAdapter<Senator> adapter = (ArrayAdapter<Senator>)
listView.getAdapter();
                int colorVal =0;
                int scaledHeight = westUSDrawable.getBounds().height();
                int scaledWidth = westUSDrawable.getBounds().width();
                int scaledImageOffsetX = Math.round(arg1.getX()) -
westUSDrawable.getBounds().left;
                int scaledImageOffsetY = Math.round(arg1.getY()) -
westUSDrawable.getBounds().top;

                int origX = (scaledImageOffsetX * westUSBitmap.getWidth() / scaledWidth);
                int origY = (scaledImageOffsetY * westUSBitmap.getHeight() /
scaledHeight);

                if(origX < 0) origX = 0;
                if(origY < 0) origY = 0;
                if(origX > westUSBitmap.getWidth()) origX = westUSBitmap.getWidth();
                if(origY > westUSBitmap.getHeight()) origY = westUSBitmap.getHeight();

                switch (arg1.getAction()) {
                case MotionEvent.ACTION_DOWN:
                        colorVal = getColour(origX, origY);
                        switch (colorVal) {
                        case -65536:
                                selectState("Pennsylvania", R.drawable.wa,
```

```
Statistics.State.Pennsylvania);
                        break;
                    case -65281:
                        selectState("Maine", R.drawable.id, Statistics.State.Maine);
                        break;
                    case -16711681:
                        selectState("New Jersey", R.drawable.or,
Statistics.State.NewJersey);
                        break;
                    case -256:
                        selectState("New York", R.drawable.ca,
Statistics.State.NewYork);
                        break;
                    case -8454144:
                        selectState("New Hampshire", R.drawable.nv,
Statistics.State.NewHampshire);
                        break;
                    case -12609351:
                        selectState("Vermont", R.drawable.ut,
Statistics.State.Vermont);
                        break;
                    case -1806524:
                        selectState("Washington", R.drawable.co,
Statistics.State.Washington);
                        break;
                    case -8665752:
                        selectState("Connecticut", R.drawable.az,
Statistics.State.Connecticut);
                        break;
                    case -2637268:
                        selectState("Rhode Island", R.drawable.nm,
Statistics.State.RhodeIsland);
                        break;
                    default:
                        break;

                    }
                    //selectedState.setVisibility(View.VISIBLE);
                    quickPane.setVisibility(View.VISIBLE);
                    adapter.notifyDataSetChanged();

                default:
                    break;
                }
                return false;
        }
```

7.2.3 State Selection Screen

Description: State selection screen displays the state that belongs to the region selected. It also specifies the divisions within the region and the states to that division.

How to get to the current screen: Tapping and holding on the region screen.

Next Action to be performed – Tapping on to a particular state will display the senator list of the state selected.

Xml code for UI – Refer Appendix B

7.2.4 Tutorial Screen

Description: The tutorial screen displays the details about the senator, policies and terms related to the senate. This screen is used to refresh the knowledge about the senator that is specified in chapter two of this document.

How to get to the current screen: Tapping on the "Tutorial" button on the main screen as part of the quick facts pane. See Figure 7.3.

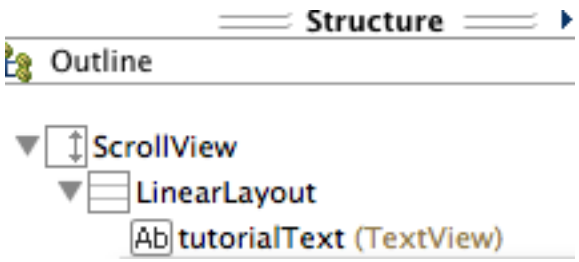


Figure 7.3. Tutorial screen layout structure.

Xml code snippet for UI – Refer Appendix B

JAVA code for Logical flow –

//TutorialActivity.java

Below code inflates the tutorialview.xml

```
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tutorialview);

        initialize();
}
```

The code below is used to initialize the widget view in the above-mentioned xml.

```
private void initialize() {

        tutorialText = (TextView) findViewById(R.id.tutorialText);
        tutorialText.setText((getResources().getString(R.string.tutorialText)));

        }
```

7.2.5 Quiz Screen

Description: The quiz screen display a screen where the user can take a quiz. This feature in the application can be a fun way to test the knowledge they have obtained using this application.

How to get to the current screen: Tapping on the "Take the quiz" button on the main screen as part of the quick facts pane. Internally the application has more than 5 collections in the quiz (Figure 7.4).

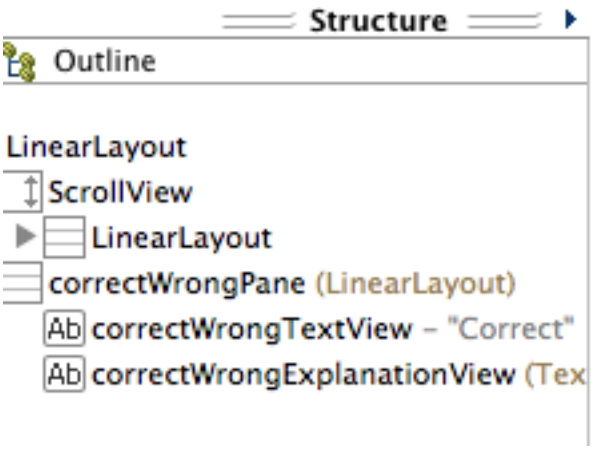


Figure 7.4. Quiz screen layout structure.

Xml code for UI – Refer Appendix B

JAVA code for Logical flow –

//QuizActivity.java

Below code inflates the quizview.xml

```
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.quizview);
```

```
        initialize();
        }
```

The code logic to perform a check to see if the chosen answer is correct/wrong is indicating the same to user is written below.

```
@Override
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        switch (group.getId()) {
        case R.id.AnswerOne:
            if(checkedId != R.id.ansOneB)
            {
                correctWrongPane.setVisibility(View.VISIBLE);
                answerText.setText("WRONG");
                answerExplanationText.setText("There are 100 senators in the
US Senate.\n Correct answer B");

            }
            else
            {
                correctWrongPane.setVisibility(View.VISIBLE);
                answerText.setText("CORRECT!!");
                answerExplanationText.setText("");


            }
            break;
        case R.id.AnswerTwo:
            if(checkedId != R.id.ansTwoD)
            {
                correctWrongPane.setVisibility(View.VISIBLE);
                answerText.setText("WRONG");
                answerExplanationText.setText("There are 2 senators per state
in the US Senate.\n Correct answer D");

            }
            else
            {
                correctWrongPane.setVisibility(View.VISIBLE);
                answerText.setText("CORRECT!!");
                answerExplanationText.setText("");


            }
            break;
        case R.id.AnswerThree:
```

```
                    if(checkedId != R.id.ansThreeC)
                    {
                            correctWrongPane.setVisibility(View.VISIBLE);
                            answerText.setText("WRONG");
                            answerExplanationText.setText("TheSenators serve in the
Senate for 6 years.\n Correct answer C");

                    }
                    else
                    {
                            correctWrongPane.setVisibility(View.VISIBLE);
                            answerText.setText("CORRECT!!");
                            answerExplanationText.setText("");


                    }
                    break;
            case R.id.AnswerFour:
                    if(checkedId != R.id.ansFourA)
                    {
                            correctWrongPane.setVisibility(View.VISIBLE);
                            answerText.setText("WRONG");
                            answerExplanationText.setText("There are two classes for
every Senate.\n Correct answer A");

                    }
                    else
                    {
                            correctWrongPane.setVisibility(View.VISIBLE);
                            answerText.setText("CORRECT!!");
                            answerExplanationText.setText("");


                    }
                    break;

            default:
                    break;
            }

    }
     }
```

7.2.6 Advanced Search Screen

Description: The advanced search screen displays the advanced feature where the user can filter the senator details based on different options on the screen. By default all the states of the U.S along with the complete list of the senators are displayed. This screen also displays a column named ‘ comments’, which indicates special circumstances during the senators elected term (if any). The different options in the advanced search feature are mentioned below:

1. Filter the list of senators based on the state.
2. Arrange the senator names in ascending or descending order using the last name.
3. Filter the list of senators based on the senate class.

How to get to the current screen: Tapping on the “Advanced search” button on the main screen as part of the quick facts pane (Figure 7.5).

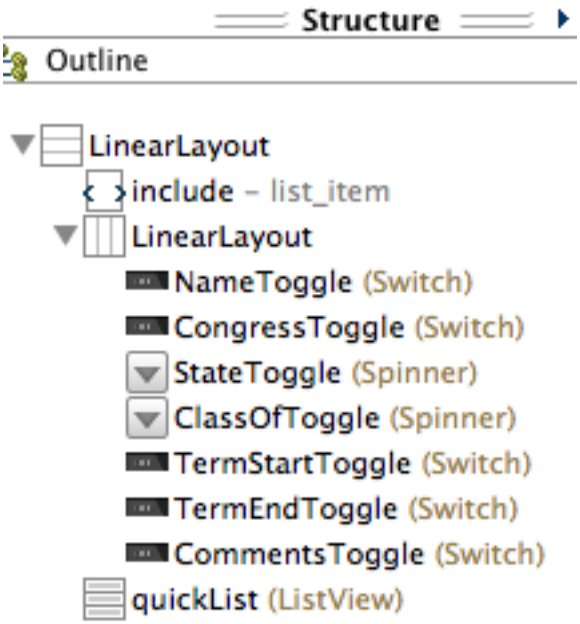


Figure 7.5. Advanced search screen layout structure.

Xml code for UI – Refer Appendix B

JAVA code for Logical flow –

Below is the code to inflate the advancedsearch.xml layout

// AdvancedSearchActivity.java

```
@Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.advancedsearch);

initialize();
}
```

CHAPTER 8

CONCULSION AND OBSTRACLES

In this chapter, the obstacles faced during the application development and an overall conclusion is discussed.

The obstacles were twofold. One, making the powerful tools of databases compatible with android and two, understanding the political context for the project.

Databases are powerful tools for data mining; searching and sorting of large sets of data. This tool is usually used in a PC environment. Adapting this tool to the emerging format of android/mobile OS presented a major challenge. Advanced tools existed to make these two disparate tools compatible. I had to make myself familiar with SQLite, which was a database plugin developed for Android.

The second challenge was to understand the US political landscape and its workings. I found a way to relate the Senate and Congress to their equivalent groups in India. This allowed me to get a better understanding of the workings of the Senate and gave me a basis for developing the app.

CHAPTER 9

FUTURE WORK

There is a great scope for extending this application project based on the student and faculty feedback.

1. Application can also have Support for State maps based on Party.
2. Application can be published in the Google play store and Amazon app store.
3. The assessment tool can be upgraded to a game like quiz tool like Quiz bowl for political science senate topic.
4. The Read me screen can be enhanced to be graphically interactive instead of reading instructions to use the application line by line (Figure 9.1).

   We could do some instruction image similar for our application.
5. Enhance the adding additional quiz questions feature to quiz tool.

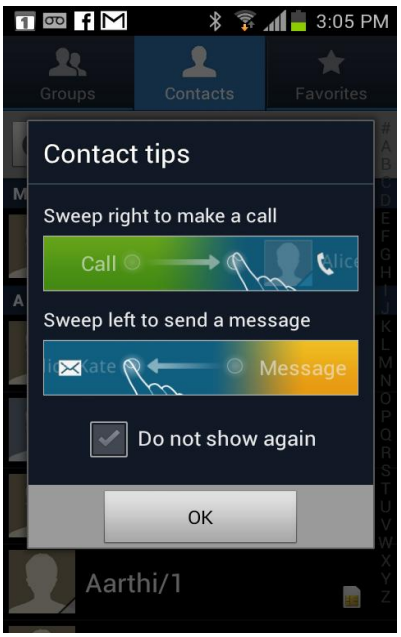


Figure 9.1. Interactive instruction sample.

REFERENCES

[1] WIKIPEDIA. *United States senate (and others).* http://en.wikipedia.org/wiki/United_States_Senate, accessed January 2013, n.d.

[2] WIKIPEDIA. *Senate*. http://en.wikipedia.org/wiki/, accessed December 2012, n.d.

[3] WIKIPEDIA. United States senate seniority. https://en.wikipedia.org/wiki/Seniority_in_the_United_States_Senate, accessed January 2013, n.d.

[4] WIKIPEDIA. *United States senate class*. https://en.wikipedia.org/wiki/Seniority_in_the_United_States_Senate, accessed January 2013, n.d.

[5] GOHARSH. *Android version and logo information*. http://www.goharsh.com/cell-phone/what-is-android/, accessed February 2013, n.d.

[6] AROKIAIT. *About Android*. http://www.arokiait.com/whatis-android.htm, accessed February 2013, n.d.

[7] WIKIPEDIA. *Android Operating system evolution details*. http://en.wikipedia.org/wiki/Android_%28operating_system%29, accessed January 2013, n.d.

[8] ANDROID DEVELOPER GUIDE. *Android emulator*. http://developer.android.com/tools/help/emulator.html, accessed January 2012, n.d.

[9] ANDROID DEVELOPER GUIDE. *Emulator tool*. http://developer.android.com/tools/devices/emulator.html, accessed February 2013, n.d.

[10] ANDROID DEVELOPER GUIDE. *Android debug bridge*. http://developer.android.com/tools/help/adb.html, accessed February 2013, n.d.

[11] ANDROID DEVELOPER GUIDE. *Android developer tool*. http://developer.android.com/tools/sdk/eclipse-adt.html, accessed February 2013, n.d.

[12] VOGELLE TUTORIAL. *SQLite Tutorial*. http://www.vogella.com/articles/AndroidSQLite/article.html, accessed February 2013, n.d.

[13] WIKIPEDIA. *United States Census bureau's list of regions*. https://en.wikipedia.org/wiki/List_of_regions_of_United_states, accessed March 2013, n.d.

[14] SENATOR GOV. *States based senator information for more than 100 years*. http://www.senate.gov/states, accessed March 2013, n.d.

APPENDIX A

USER MANUAL

Follow the below instruction to set up the development environment for android application.

Initial Steps:

1. Download an IDE –

   We could use any IDE for the application development. I have used Eclipse as the IDE for U.S senator application development. Download the latest version of the software here: http://www.eclipse.org/downloads/packages/release/indigo/r. Before downloading the IDE choose the version that corresponds to your OS of choice.

2. Install Eclipse –

   Make sure to take note of the installation directory and workspace for later reference.

3. Download and install the Java Runtime Environment (JRE) –

   Make sure to download JRE if you don't have it installed already. User can find the download file here: http://www.oracle.com/technetwork/java/javase/downloads/jre-6u25-download-346243.html

4. Download and install the Java Development Kit –

   If the user don't have Java installed in the development PC. They can download the file from the link below:

   http://www.oracle.com/technetwork/java/javase/downloads/jdk-7u3-download-1501626.html

Android SDK installation steps –

1. Install the Android SDK –

   The SDK starter package includes the Android SDK and AVD Manager. This is a tool could be used to download some required SDK components into our development environment. Flow the screen shot below for reference.

Figure A.1. Screen shot of SDK starter package includes the Android SDK and AVD Manager.

Downloaded starter package only contains the newest version of the SDK tools. In order to develop Android applications, we will need to download at least one Android platform so we can emulate a phone/tablet via our computer hardware.

2. In the root of the SDK directory, find and execute the SDK Manager.exe file. Note: Right-click and run as administrator to avoid any permission declination.

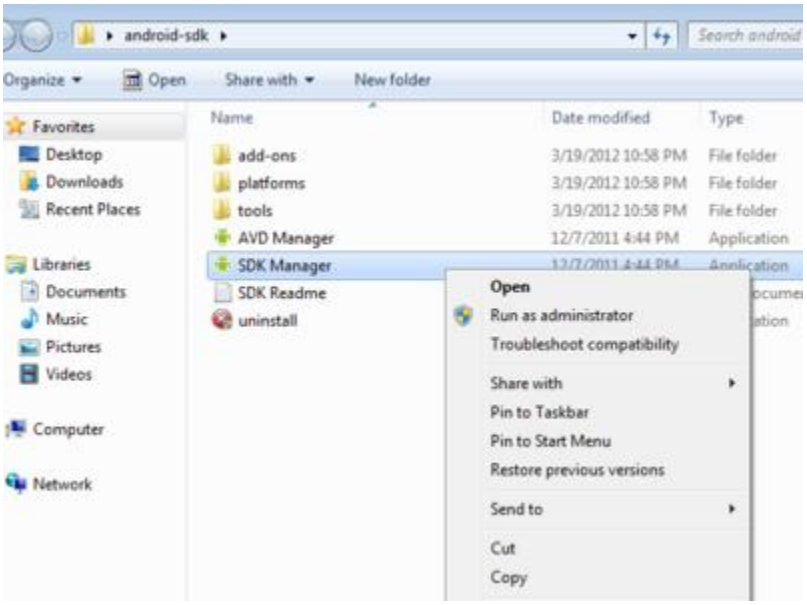


Figure A.2. Steps to find and execute SDK manager.

Choose the Android platform version, which user wishes to develop on (For our application purpose it would be beneficial to download Android platform 2.3 (API 9) to Android 4.1 (API 16). For further information on android versions refer chapter 3) and click Install packages. You will be prompted with a pop-up, check off Accept and click Install.

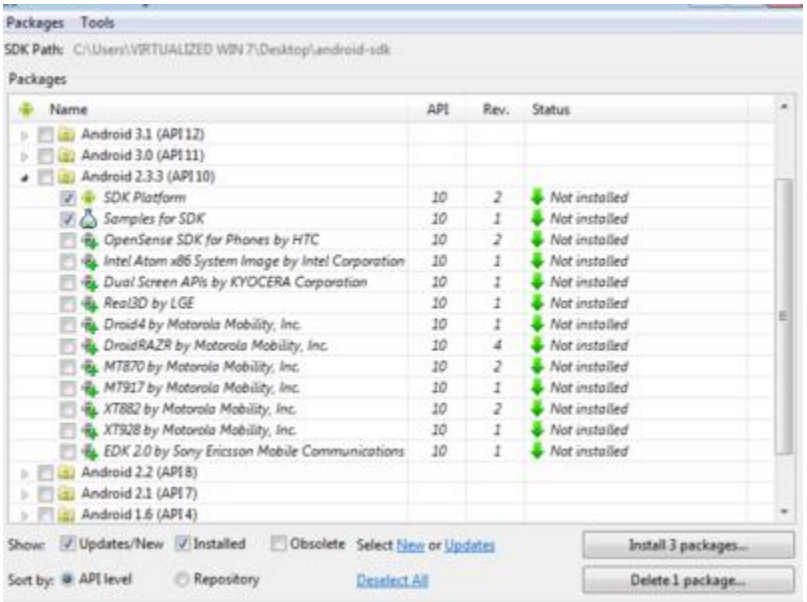

Packages
Tools
SDK Path: C:\Users\VIRTUALIZED WIN 7\Desktop\android-sdk
Packages
Name
API
Rev.
Status
Android 3.1 (API 12)
Android 3.0 (API 11)
Android 2.3.3 (API 10)
SDK Platform 10 2 Not installed
Samples for SDK 10 1 Not installed
OpenSense SDK for Phones by HTC 10 2 Not installed
Intel Atom x86 System Image by Intel Corporation 10 1 Not installed
Dual Screen APIs by KYOCERA Corporation 10 1 Not installed
Real3D by LGE 10 1 Not installed
Droid4 by Motorola Mobility, Inc. 10 1 Not installed
DroidRAZR by Motorola Mobility, Inc. 10 4 Not installed
MT870 by Motorola Mobility, Inc. 10 2 Not installed
MT917 by Motorola Mobility, Inc. 10 1 Not installed
XT882 by Motorola Mobility, Inc. 10 2 Not installed
XT928 by Motorola Mobility, Inc. 10 1 Not installed
EDK 2.0 by Sony Ericsson Mobile Communications 10 1 Not installed
Android 2.2 (API 8)
Android 2.1 (API 7)
Android 1.6 (API 4)
Show: Updates/New Installed Obsolete Select New or Updates
Install 3 packages...
Sort by: API level Repository
Deselect All
Delete 1 package...

Figure A.3. Android platform version choice.

3. Download and Install the Android Development Tools (ADT) plugin for Eclipse – Google provides a custom plugin for Eclipse, called ADT, that aids developers in creating Android applications faster and more efficiently. Open Eclipse and select Help-->Install New Software

   Figure A4, show the screen shot of how to Open Eclipse and select Help-->Install

   New Software.

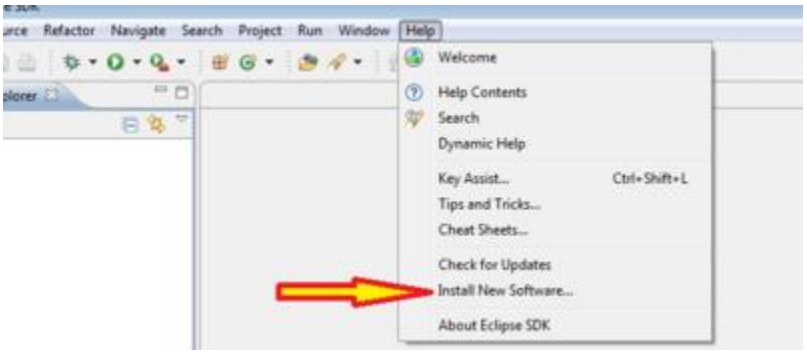


Figure A.4. Screen shot of how to open Eclipse.

Then click Add option. In the pop-up dialog, type 'ADT Plugin' in the Name field and enter the below URL in the location field:

https://dl-ssl.google.com/android/eclipse

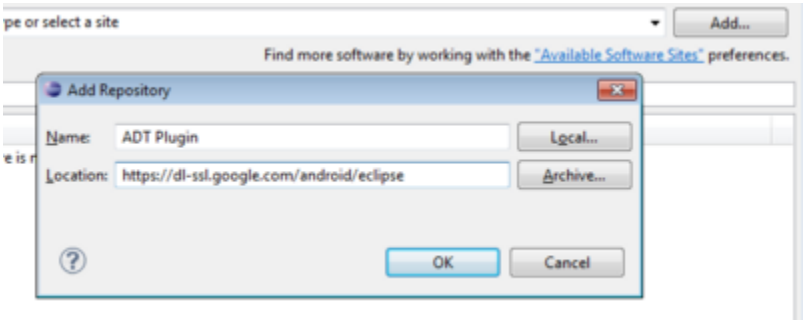


Figure A.5. Screen shot of the dialog to type 'ADT Plugin'.

Figure A5 , show the screen shot of the dialog to type 'ADT Plugin' in the Name field and enter the following URL in the location field

Then Click OK. If you're having issues in downloading the plugin, use "http" instead of "https". Select the 'Developer Tools' option and click Next. Once the tools are downloaded, click Next.

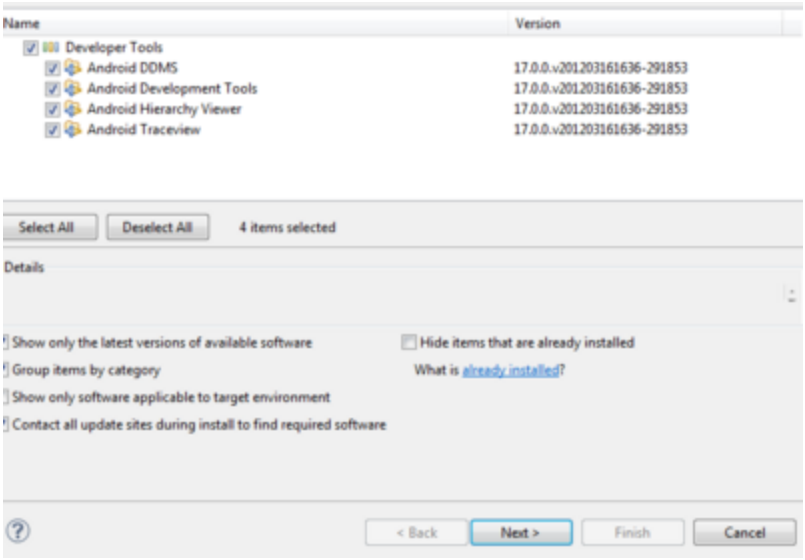


Figure A.6. Screenshot of 'developer tools' option.

Figure A6, screenshot of 'developer tools' option

Finally, Read and accept the License Agreements. When you are done, click Finish and restart Eclipse. Note: you may encounter a Security Warning message notifying you of unsigned content. If you do, click OK.

4. **Configure ADT plugin**

   In Eclipse, select Window --> Preferences. Then select Android from the panel on the left. Choose if you want to opt into sending usage statistics and click Proceed.

   Figure A7, screenshot to select Android from the panel on the left.

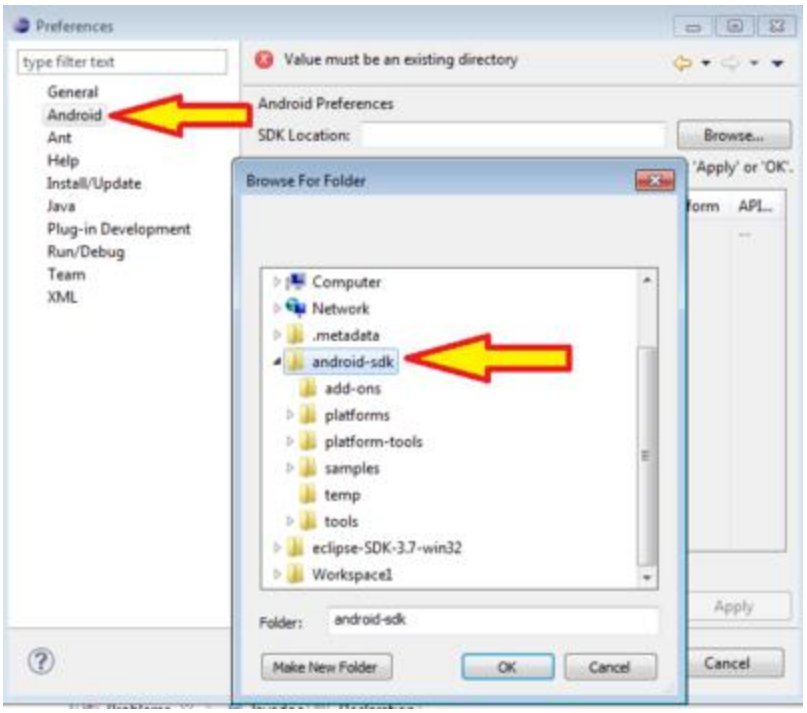


Figure A.7. Screenshot to select Android from the panel on the left.

In the SDK Location field, click Browse and choose the directory where you saved the Android SDK in step. Then Click Apply -->OK. You have now configured the ADT plugin successfully.

5. Finally **Configure an AVD**

   set up an AVD so that you can emulate a phone/tablet using your computer hardware. This will allow you to test the application being developed without requiring a physical Android device.

   In Eclipse, navigate to Window --> AVD Manager. Click New… to fill in the details of the virtual device.

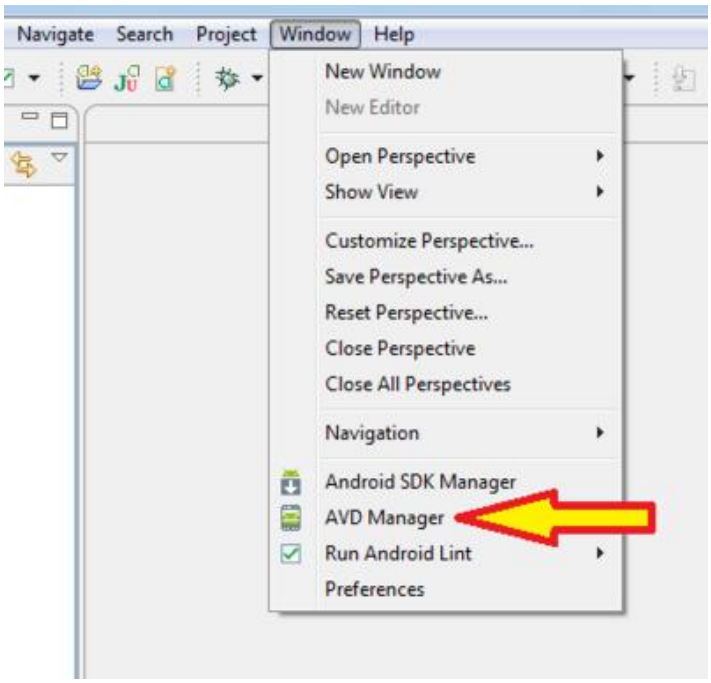


Figure A.8. Screen shot in Eclipse, navigate to Window --> AVD Manager.

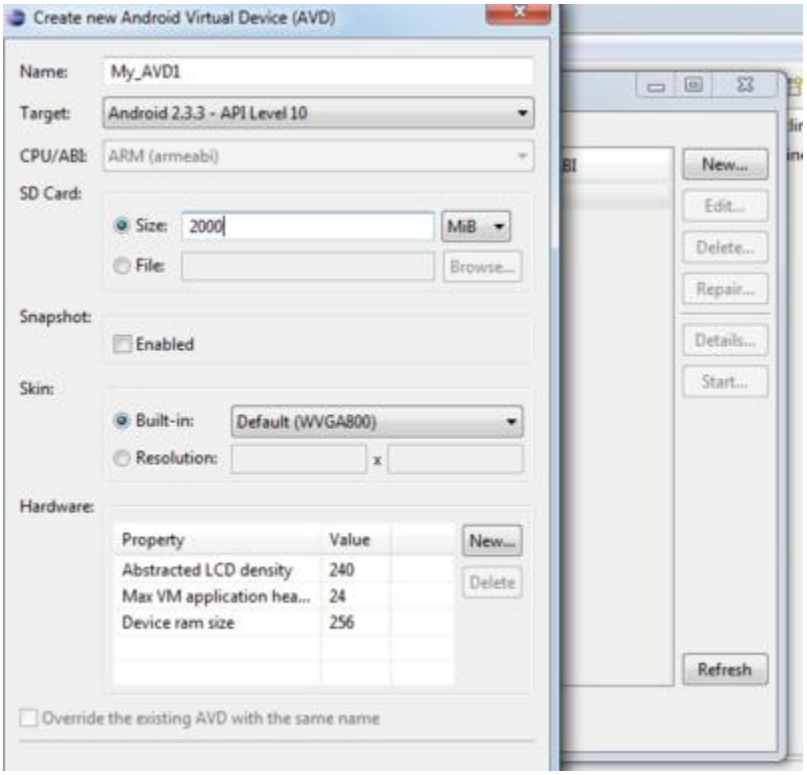


Figure A.9. Screenshot how to create AVD.

Choose a name that conforms to the naming rules specified at the bottom of the window. The Target field specifies the version of Android you wish to have the

virtual device running on. Notice that the list of the drop down menu only contains version of the Android OS that you selected to download during the SDK installation step. If you want the AVD to run another version of the OS, you must download appropriate version packages in the SDK Manager.

The rest of the options relate to the specific features that you wish to customize the AVD with. These features represent hardware features present on physical Android devices. For instance, you may customize the screen resolution of the emulated device, availability of GPS. You can even specify a custom battery level state to test particular usage scenarios that app-users may encounter.

Once the user is satisfied with the configuration of the AVD, click Create AVD.

Steps for running the application:

a. In your project in eclipse go to Run Configurations

Figure A.10. Show the screenshot of run configuration option.

b. In the target tabchoose the option: “Always prompt to pick the device” and click Run.

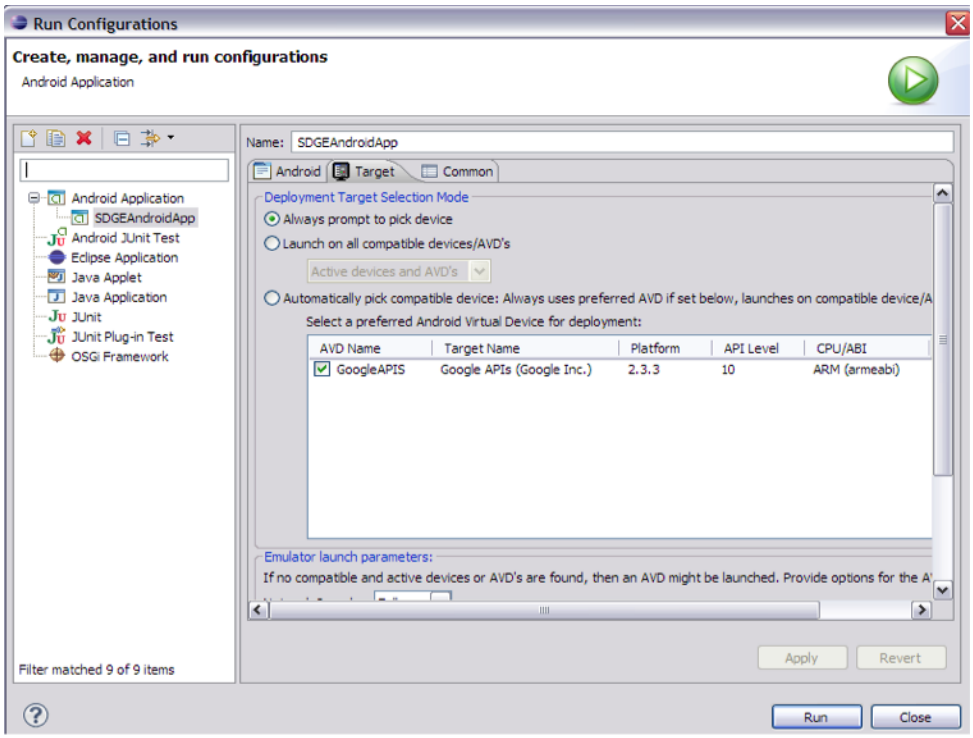


Figure A.11. Shows the screenshot of emulator/device option.

How to Run on Emulator –

Every time you run the application, android will prompt you to choose a running device/emulator option. The AVD you created in AVD Manager will be listed as shown above and can be used to run the application.

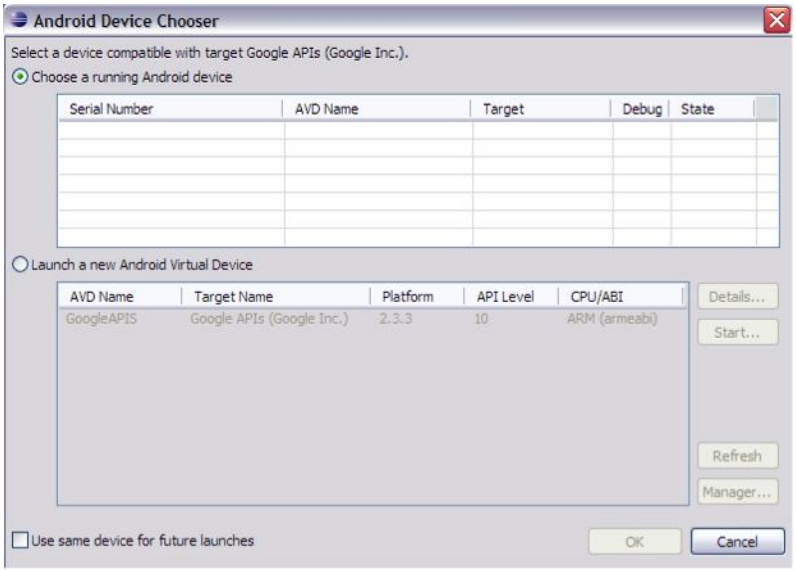


Figure A.12. Shows the screen shot of emulator option.

How to Run on Device –

To run the application on your device, you will have to install the device driver. Then, connect the device to the system and it will be listed in the Android Device Chooser window.

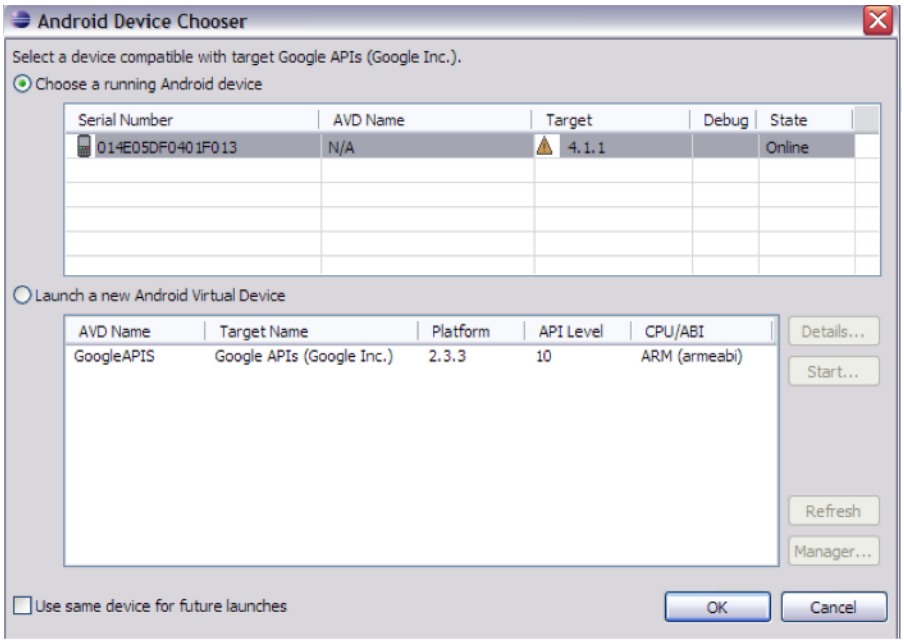


Figure A.13. Screen shot to run on device.

APPENDIX B

APPLICATION CODE

This section discusses different XML and JAVA files that are used in developing this application.

When the application loads the US map loads with current party strength. Below is the XML code and graphical layout that was developed for populating it.

//main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >

   <LinearLayout
      android:layout_width="fill_parent"
      android:layout_height="0dp"
      android:layout_weight="9"
      android:orientation="horizontal" >

      <RelativeLayout
         android:layout_width="0dp"
         android:layout_height="wrap_content"
         android:layout_weight="4"
         android:orientation="vertical" >

         <ImageView
            android:id="@+id/usmap"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:background="@drawable/blankusmapgrey"
            android:visibility="visible" />

         <ImageView
            android:id="@+id/regionsusmap"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:background="@drawable/regions_us_map"
            android:visibility="invisible" />

         <ImageView
            android:id="@+id/activeregion"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:visibility="invisible" />
      </RelativeLayout>
```

```
<LinearLayout
   android:id="@+id/quickPane"
   android:layout_width="0dp"
   android:layout_height="fill_parent"
   android:layout_weight="1"
   android:background="@color/LightBlue"
   android:orientation="vertical"
   android:visibility="gone" >

   <LinearLayout
      android:layout_width="fill_parent"
      android:layout_height="0dp"
      android:layout_weight="4"
      android:orientation="vertical" >

      <TextView
         android:id="@+id/regionName"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_gravity="center"
         android:text="Senators of USA"
         android:textAppearance="?android:attr/textAppearanceLarge" />

      <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_gravity="center"
         android:layout_marginTop="10dp"
         android:text="Quick Facts"
         android:textAppearance="?android:attr/textAppearanceSmall" />

      <View
         android:layout_width="fill_parent"
         android:layout_height="3dp"
         android:background="@color/Black" />


      <ListView
         android:id="@+id/quickList"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:layout_marginTop="3dp">
      </ListView>
   </LinearLayout>
```

```
        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="0dp"
            android:layout_weight="2"
            android:gravity="bottom"
            android:orientation="vertical" >

            <Button
                android:id="@+id/cancelButton"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="bottom"
                android:text="Tutorial"
                android:textSize="@dimen/quick_pane_text_size" />

            <Button
                android:id="@+id/quizButton"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="bottom"
                android:text="Take a Quiz"
                android:textSize="@dimen/quick_pane_text_size" />

            <Button
                android:id="@+id/moreButton"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="bottom"
                android:text="Advanced Search "
                android:textSize="@dimen/quick_pane_text_size" />
        </LinearLayout>
    </LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:orientation="horizontal" >

    <TextView
        android:id="@+id/repNum"
        android:layout_width="0dp"
        android:layout_height="fill_parent"
```

```
        android:layout_gravity="bottom"
        android:layout_weight="1"
        android:background="@color/Blue"
        android:gravity="center"
        android:text="Rep: "
        android:textSize="@dimen/rep_dem_ind_size" />

    <TextView
        android:id="@+id/demNum"
        android:layout_width="0dp"
        android:layout_height="fill_parent"
        android:layout_gravity="bottom"
        android:layout_weight="1"
        android:background="@color/Red"
        android:gravity="center"
        android:text="Dem: "
        android:textSize="@dimen/rep_dem_ind_size" />

    <TextView
        android:id="@+id/indNum"
        android:layout_width="0dp"
        android:layout_height="fill_parent"
        android:layout_gravity="bottom"
        android:layout_weight="1"
        android:background="@color/Gray"
        android:gravity="center"
        android:text="Ind: "
        android:textSize="@dimen/rep_dem_ind_size" />
</LinearLayout>

    </LinearLayout>
```

Graphical representation –

Below is the Graphical representation of US map xml in eclipse.

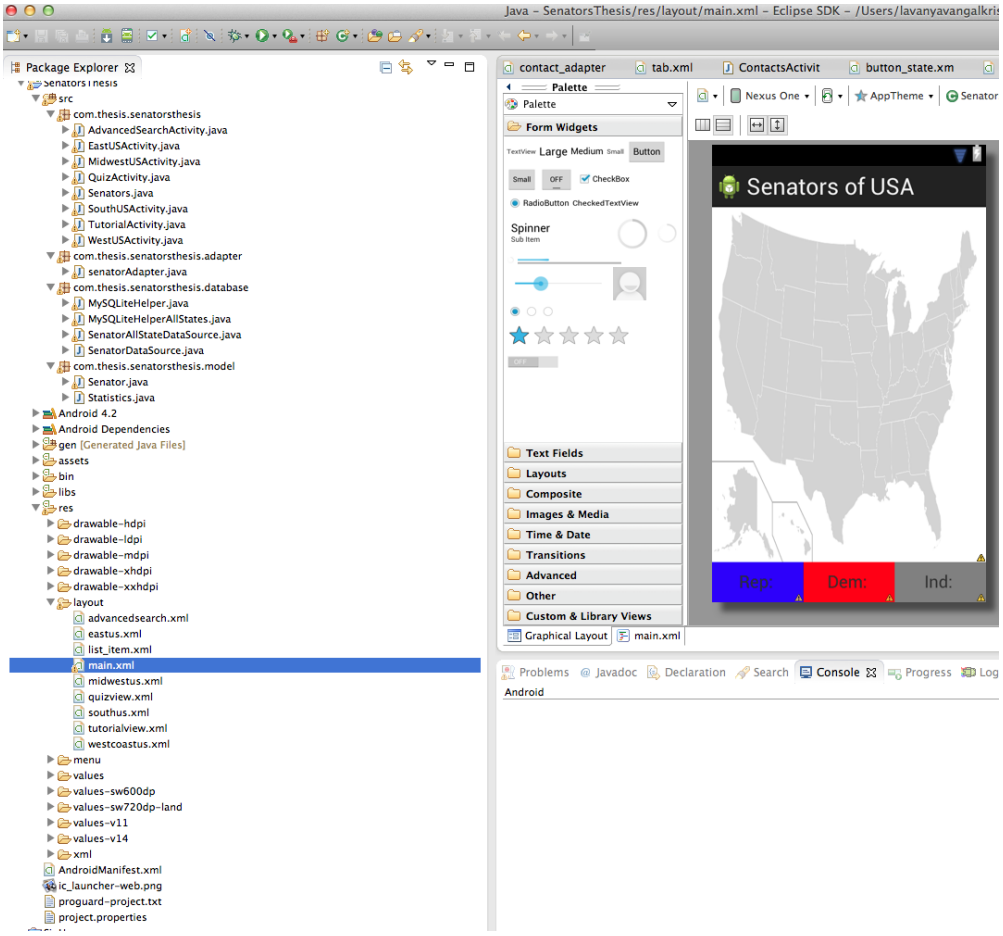


Figure B.1. Shows main.xml graphical layout.

Below is the List of all the XML Files along with the Graphical representation of each region creation for this application.

Westcoastus.xml - This xml is used to create the map of all the western states in US with their on click action listeners.

// westcoastus.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@color/LightYellow"
   android:orientation="horizontal" >

   <RelativeLayout
      android:layout_width="fill_parent"
```

```
android:layout_height="match_parent"
android:orientation="vertical" >

<ImageView
   android:id="@+id/westusmap"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:background="@drawable/west_map"
   android:visibility="visible" />

<ImageView
   android:id="@+id/westusstates"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:background="@drawable/west_colormap"
   android:visibility="invisible" />

<ImageView
   android:id="@+id/selectedstate"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:alpha="20"
   android:visibility="invisible" />

<LinearLayout
android:id="@+id/quickPane"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:background="@color/LightBlue"
android:orientation="vertical"
android:visibility="gone" >

<LinearLayout
   android:layout_width="fill_parent"
   android:layout_height="0dp"
   android:layout_weight="4"
   android:orientation="vertical" >

   <TextView
      android:id="@+id/regionName"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center"
      android:text="Senators of USA"
      android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="10dp"
        android:text="Quick Facts"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <View
        android:layout_width="fill_parent"
        android:layout_height="3dp"
        android:background="@color/Black" />

    <include layout = "@layout/list_item"/>

    <ListView
        android:id="@+id/quickList"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="3dp" >
    </ListView>
</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:gravity="bottom"
    android:orientation="vertical" >

    <Button
        android:id="@+id/cancelButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:text="Hide "
        android:textSize="@dimen/quick_pane_text_size" />

    <Button
        android:id="@+id/moreButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:text="More "
        android:textSize="@dimen/quick_pane_text_size" />
</LinearLayout>
```

```
</LinearLayout>

</RelativeLayout>

    </LinearLayout>
```

Graphical representation –

Below is the Graphical representation of westcoastus states xml in eclipse.

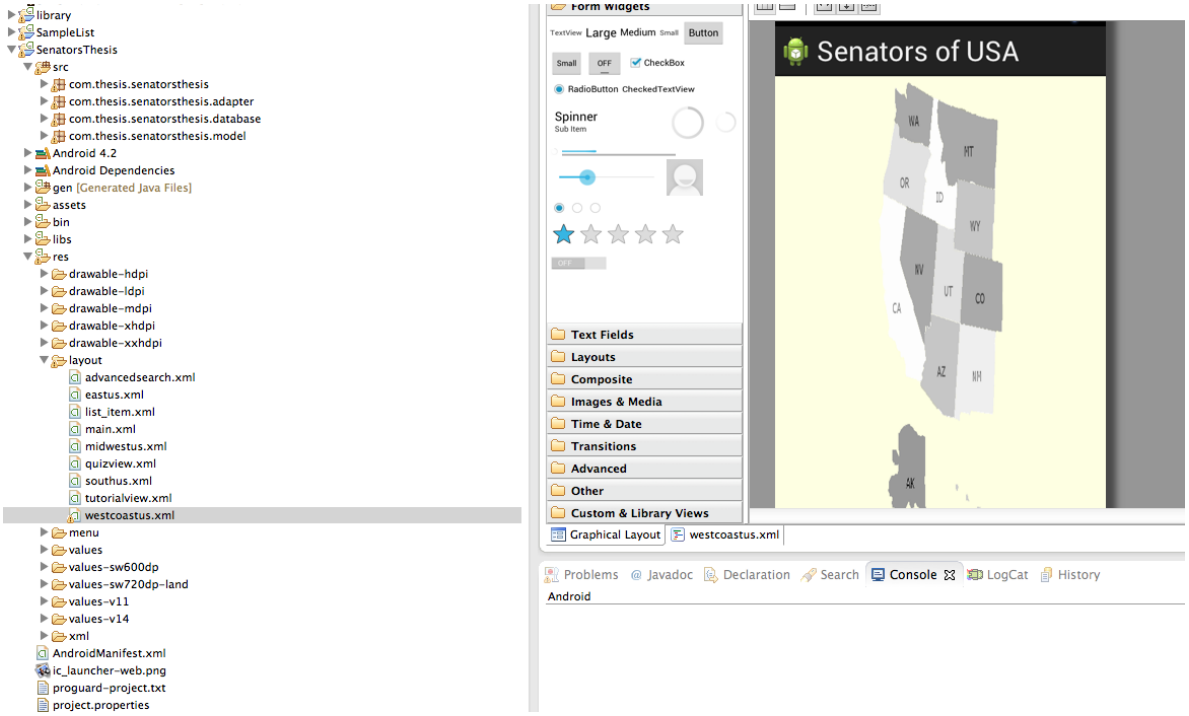


Figure B.2. Shows westcoastus.xml graphical layout.

eastus.xml - This xml is used to create the map of all the eastern states in US with their on click action listeners.

// eastus.xml -

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@color/LightYellow"
   android:orientation="horizontal" >

   <RelativeLayout
```

```
android:layout_width="fill_parent"
android:layout_height="match_parent"
android:orientation="vertical" >

<ImageView
   android:id="@+id/westusmap"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:background="@drawable/east_map"
   android:visibility="visible" />

<ImageView
   android:id="@+id/westusstates"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:background="@drawable/east_colormap"
   android:visibility="invisible" />

<ImageView
   android:id="@+id/selectedstate"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:alpha="20"
   android:visibility="invisible" />

<LinearLayout
android:id="@+id/quickPane"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:background="@color/LightBlue"
android:orientation="vertical"
android:visibility="gone" >

<LinearLayout
   android:layout_width="fill_parent"
   android:layout_height="0dp"
   android:layout_weight="4"
   android:orientation="vertical" >

   <TextView
      android:id="@+id/regionName"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center"
      android:text="Senators of USA"
      android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="10dp"
        android:text="Quick Facts"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <View
        android:layout_width="fill_parent"
        android:layout_height="3dp"
        android:background="@color/Black" />

    <include layout = "@layout/list_item"/>

    <ListView
        android:id="@+id/quickList"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="3dp" >
    </ListView>
</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:gravity="bottom"
    android:orientation="vertical" >

    <Button
        android:id="@+id/cancelButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:text="Hide "
        android:textSize="@dimen/quick_pane_text_size" />

    <Button
        android:id="@+id/moreButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:text="More "
        android:textSize="@dimen/quick_pane_text_size" />
```

```
    </LinearLayout>
</LinearLayout>

</RelativeLayout>


    </LinearLayout>
```

Graphical representation –

Below is the Graphical representation of eastus state xml in eclipse.

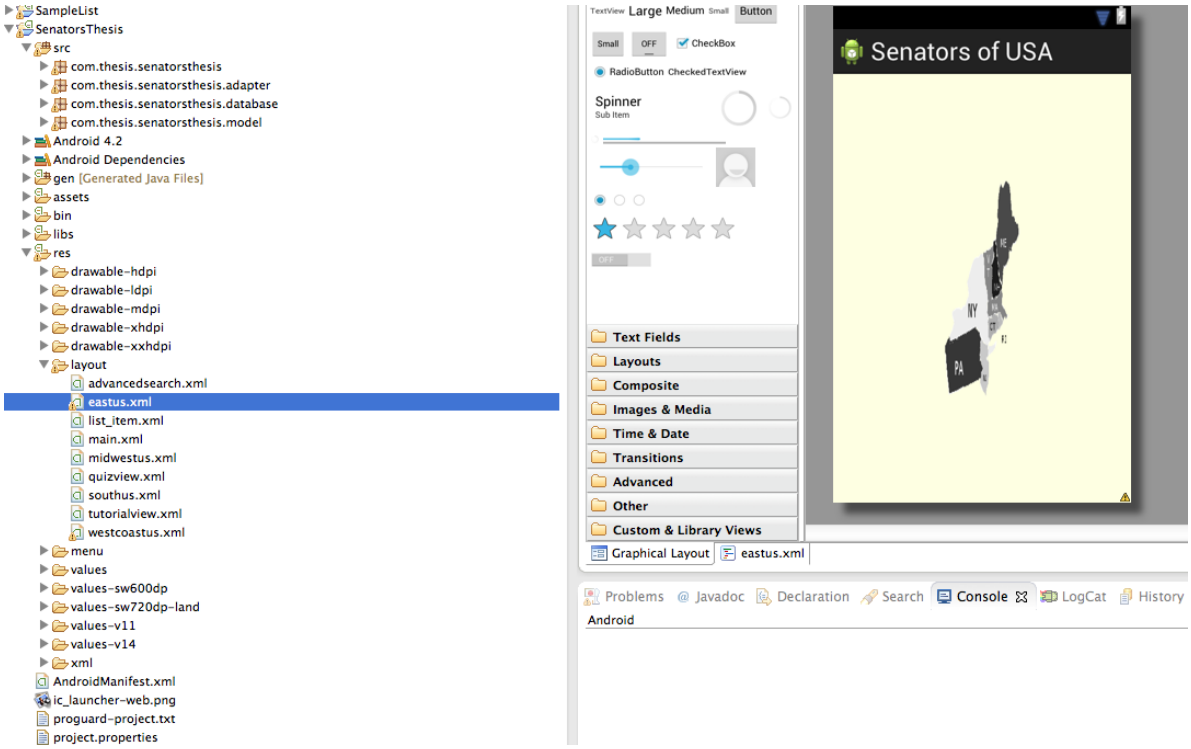


Figure B.3. Shows estus.xml graphical layout.

southus.xml - This xml is used to create the map of all the southern states in US with their on click action listeners.

// southus.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@color/LightYellow"
   android:orientation="horizontal" >
```

```
<RelativeLayout
   android:layout_width="fill_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >

   <ImageView
      android:id="@+id/westusmap"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:background="@drawable/south_map"
      android:visibility="visible" />

   <ImageView
      android:id="@+id/westusstates"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:background="@drawable/south_colormap"
      android:visibility="invisible" />

   <ImageView
      android:id="@+id/selectedstate"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:alpha="20"
      android:visibility="invisible" />

   <LinearLayout
   android:id="@+id/quickPane"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:background="@color/LightBlue"
   android:orientation="vertical"
   android:visibility="gone" >

   <LinearLayout
      android:layout_width="fill_parent"
      android:layout_height="0dp"
      android:layout_weight="4"
      android:orientation="vertical" >

      <TextView
         android:id="@+id/regionName"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_gravity="center"
```

```
        android:text="Senators of USA"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="10dp"
        android:text="Quick Facts"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <View
        android:layout_width="fill_parent"
        android:layout_height="3dp"
        android:background="@color/Black" />

    <include layout = "@layout/list_item"/>

    <ListView
        android:id="@+id/quickList"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="3dp" >
    </ListView>
</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:gravity="bottom"
    android:orientation="vertical" >

    <Button
        android:id="@+id/cancelButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:text="Hide "
        android:textSize="@dimen/quick_pane_text_size" />

    <Button
        android:id="@+id/moreButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
```

```
            android:text="More "
            android:textSize="@dimen/quick_pane_text_size" />
        </LinearLayout>
    </LinearLayout>

    </RelativeLayout>


        </LinearLayout>
```

Graphical representation –

Below is the Graphical representation of southern states xml in eclipse.

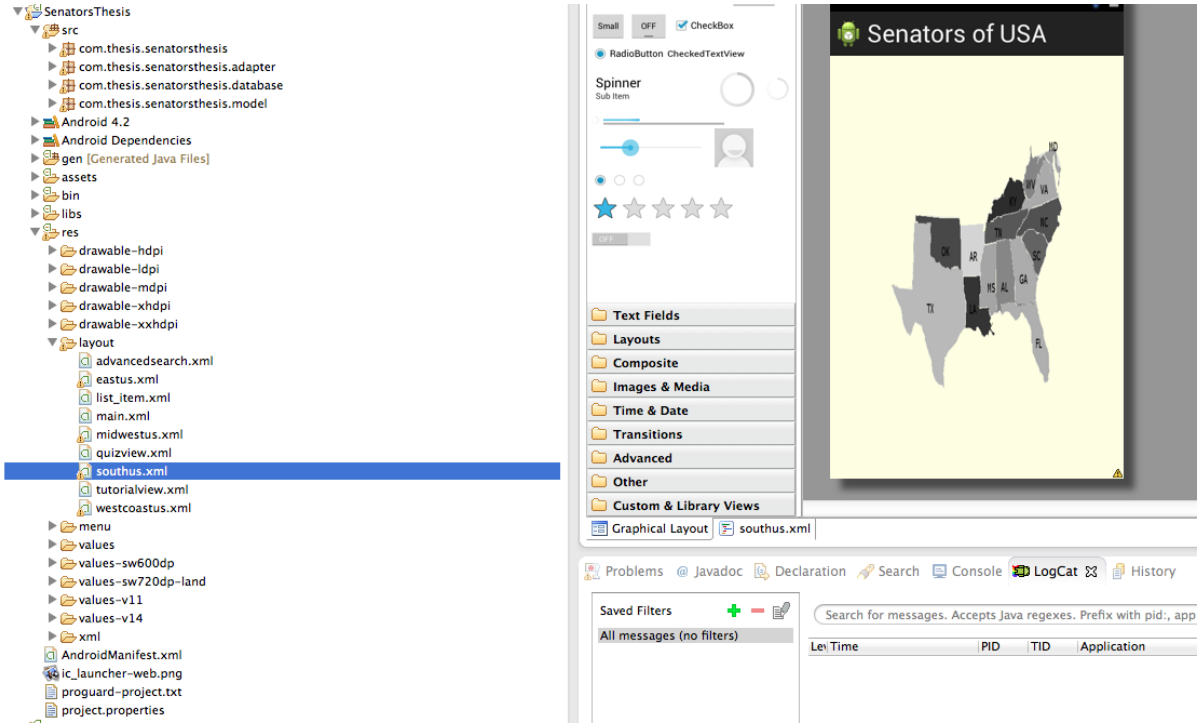


Figure B.4. Shows southus.xml graphical layout.

midwestus.xml - This xml is used to create the map of all the mid-west states in US with their on click action listener

// midwestus.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
android:background="@color/LightYellow"
android:orientation="horizontal" >

<RelativeLayout
   android:layout_width="fill_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >

   <ImageView
      android:id="@+id/westusmap"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:background="@drawable/midwest_map"
      android:visibility="visible" />

   <ImageView
      android:id="@+id/westusstates"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:background="@drawable/midwest_colormap"
      android:visibility="invisible" />

   <ImageView
      android:id="@+id/selectedstate"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:alpha="20"
      android:visibility="invisible" />

   <LinearLayout
   android:id="@+id/quickPane"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:background="@color/LightBlue"
   android:orientation="vertical"
   android:visibility="gone" >

   <LinearLayout
      android:layout_width="fill_parent"
      android:layout_height="0dp"
      android:layout_weight="4"
      android:orientation="vertical" >

      <TextView
         android:id="@+id/regionName"
         android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Senators of USA"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="10dp"
        android:text="Quick Facts"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <View
        android:layout_width="fill_parent"
        android:layout_height="3dp"
        android:background="@color/Black" />

    <include layout = "@layout/list_item"/>

    <ListView
        android:id="@+id/quickList"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="3dp" >
    </ListView>
</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:gravity="bottom"
    android:orientation="vertical" >

    <Button
        android:id="@+id/cancelButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:text="Hide "
        android:textSize="@dimen/quick_pane_text_size" />

    <Button
        android:id="@+id/moreButton"
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:text="More "
        android:textSize="@dimen/quick_pane_text_size" />
    </LinearLayout>
  </LinearLayout>

  </RelativeLayout>


    </LinearLayout>
```

Graphical representation –

Below is the Graphical representation of Mid-west states xml in eclipse.

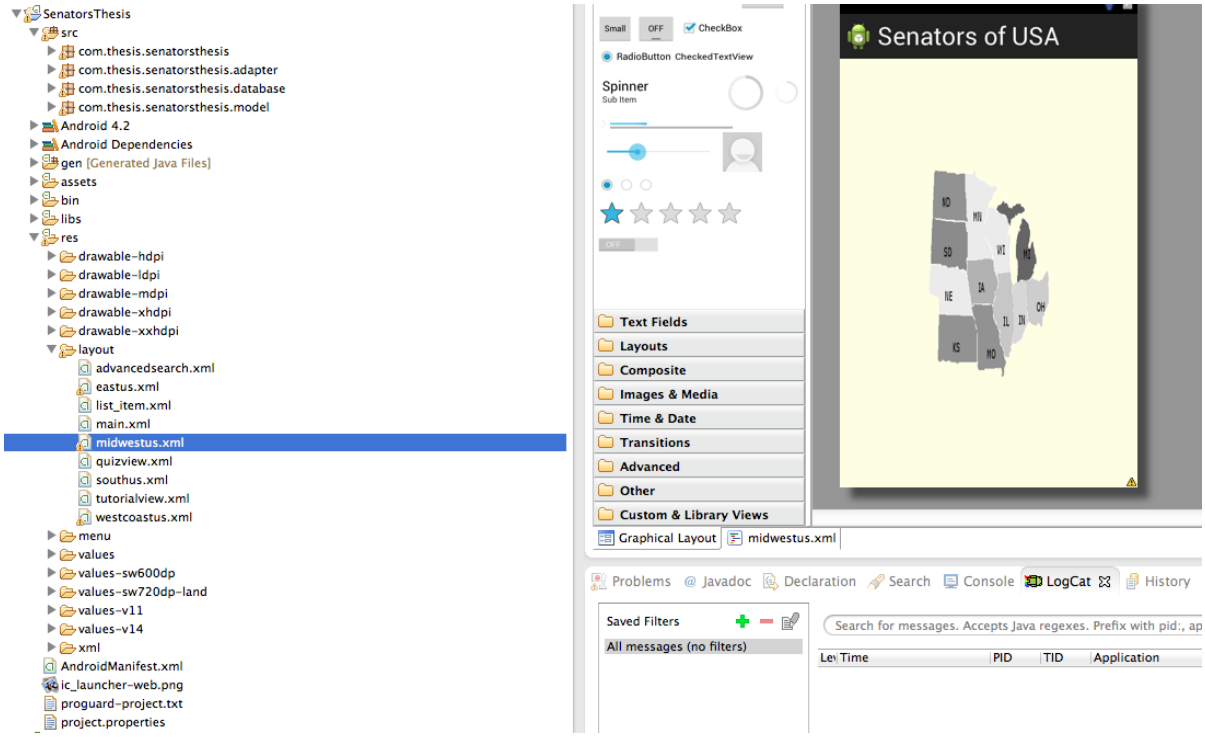


Figure B.5. Shows midwestus.xml graphical layout.

The ‘quiz’ features XML layout is displayed in the below section with the XML code and graphical layout.

// quizview.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical" >

<ScrollView
   android:layout_width="match_parent"
   android:layout_height="0dp"
   android:layout_weight="3" >

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:orientation="vertical" >

      <TextView
         android:id="@+id/QuestionOne"
         android:layout_width="match_parent"
         android:layout_height="match_parent"
         android:text="1) How many senators serve in the US senate?" />

      <RadioGroup
         android:id="@+id/AnswerOne"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content" >

         <RadioButton
            android:id="@+id/ansOneA"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:checked="true"
            android:text="107" />

         <RadioButton
            android:id="@+id/ansOneB"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="100" />

         <RadioButton
            android:id="@+id/ansOneC"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="107" />

         <RadioButton
            android:id="@+id/ansOneD"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="97" />
</RadioGroup>

<TextView
    android:id="@+id/QuestionTwo"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="2) How many senators serve from every State?" />

<RadioGroup
    android:id="@+id/AnswerTwo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
        android:id="@+id/ansTwoeA"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="3" />

    <RadioButton
        android:id="@+id/ansTwoB"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Depends on the state Size" />

    <RadioButton
        android:id="@+id/ansTwoC"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1" />

    <RadioButton
        android:id="@+id/ansTwoD"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="2" />
</RadioGroup>

<TextView
    android:id="@+id/QuestionThree"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
    android:text="3) How many years do the Senators serve in the Senate?" />

<RadioGroup
    android:id="@+id/AnswerThree"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
        android:id="@+id/ansThreeA"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="3 yrs" />

    <RadioButton
        android:id="@+id/ansThreeB"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="4 yrs" />

    <RadioButton
        android:id="@+id/ansThreeC"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="6 yrs" />

    <RadioButton
        android:id="@+id/ansThreeD"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Lifelong" />
</RadioGroup>

<TextView
    android:id="@+id/QuestionFour"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="4) How many classes from every state?" />

<RadioGroup
    android:id="@+id/AnswerFour"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
        android:id="@+id/ansFourA"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:checked="true"
            android:text="2" />

        <RadioButton
            android:id="@+id/ansFourB"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="4" />

        <RadioButton
            android:id="@+id/ansFourC"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="1" />

        <RadioButton
            android:id="@+id/ansFourD"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="3" />
    </RadioGroup>
  </LinearLayout>
</ScrollView>

<LinearLayout
  android:id="@+id/correctWrongPane"
  android:layout_width="match_parent"
  android:layout_height="0dp"
  android:layout_weight="1"
  android:background="@color/LightBlue"
  android:orientation="vertical" >
  <TextView
    android:id="@+id/correctWrongTextView"
    android:textSize="30dp"
    android:text="Correct"
    android:layout_gravity="center_horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
   <TextView
    android:id="@+id/correctWrongExplanationView"
    android:text="Explanation text here"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
</LinearLayout>
```

</LinearLayout>

Graphical representation –

Below is the Graphical representation of quiz xml in eclipse.

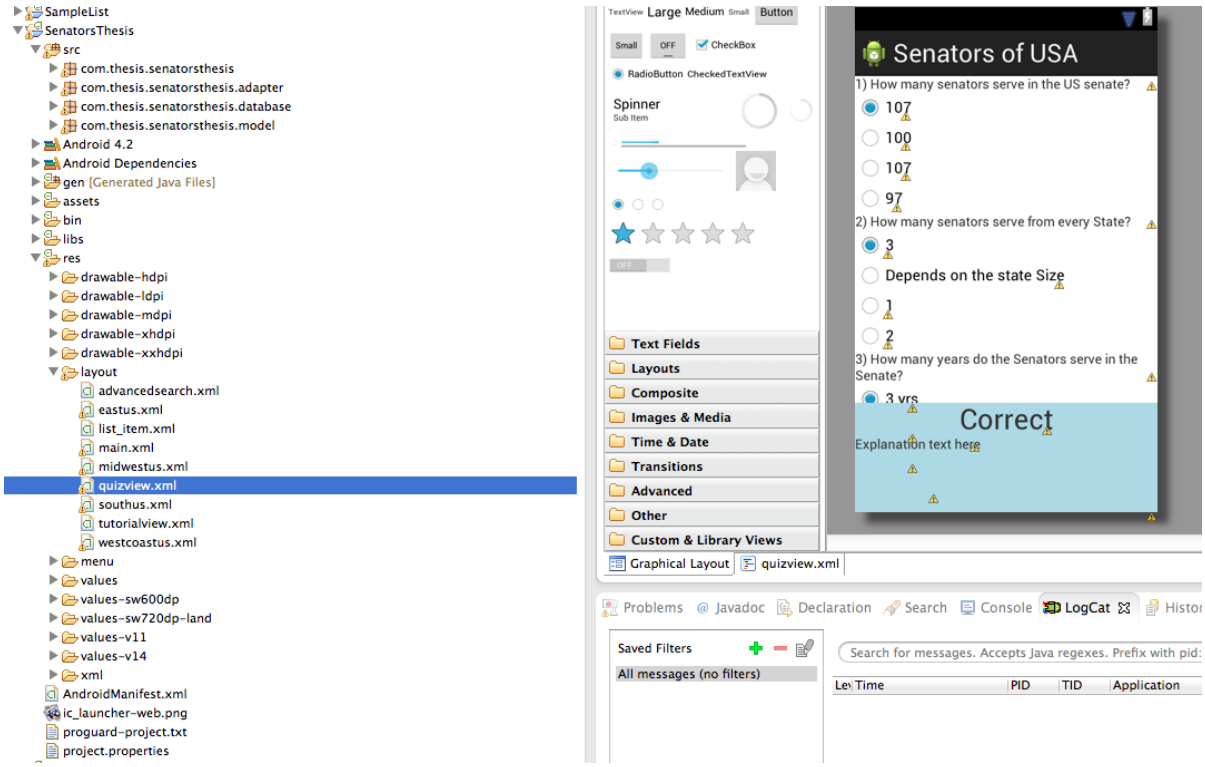


Figure B.6. Shows quizview.xml graphical layout.